Proceedings of the 14th ISWC workshop on Ontology Matching (OM)

Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Cassia Trojahn dos Santos

Ontology Matching

OM-2019

Proceedings of the ISWC Workshop

Introduction

Ontology matching[1] is a key interoperability enabler for the semantic web, as well as a useful tactic in some classical data integration tasks dealing with the semantic heterogeneity problem. It takes ontologies as input and determines as output an alignment, that is, a set of correspondences between the semantically related entities of those ontologies. These correspondences can be used for various tasks, such as ontology merging, data translation, query answering or navigation over knowledge graphs. Thus, matching ontologies enables the knowledge and data expressed with the matched ontologies to interoperate.

The workshop had three goals:

- To bring together leaders from *academia*, *industry* and *user institutions* to assess how academic advances are addressing real-world requirements. The workshop strives to improve academic awareness of industrial and final user needs, and therefore, direct research towards those needs. Simultaneously, the workshop serves to inform industry and user representatives about existing research efforts that may meet their requirements. The workshop also investigated how the ontology matching technology is going to evolve.
- To conduct an extensive and rigorous evaluation of ontology matching and instance matching (link discovery) approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2019 campaign[2].
- To examine similarities and differences from other, old, new and emerging, techniques and usages, such as process matching, web table matching or knowledge embeddings.

The program committee selected 3 long and 2 short submissions for oral presentation and 7 submissions for poster presentation. 20 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2019.ontologymatching.org/`.

[1] `http://www.ontologymatching.org/`
[2] `http://oaei.ontologymatching.org/2019`

Acknowledgments. We thank all members of the program committee, authors and local organizers for their efforts. We appreciate support from the Trentino as a Lab[3] initiative of the European Network of the Living Labs[4] at Trentino Digitale[5], the EU SEALS (Semantic Evaluation at Large Scale) project[6], the EU HOBBIT (Holistic Benchmarking of Big Linked Data) project[7], the Pistoia Alliance Ontologies Mapping project[8] and IBM Research[9].

Pavel Shvaiko
Jérôme Euzenat
Ernesto Jiménez-Ruiz
Oktie Hassanzadeh
Cássia Trojahn

December 2019

[3] http://www.taslab.eu
[4] http://www.openlivinglabs.eu
[5] http://www.trentinodigitale.it
[6] http://www.seals-project.eu
[7] https://project-hobbit.eu/challenges/om2019/
[8] http://www.pistoiaalliance.org/projects/ontologies-mapping/
[9] research.ibm.com

Organization

Organizing Committee

Pavel Shvaiko,
Trentino Digitale SpA, Italy

Jérôme Euzenat,
INRIA & University Grenoble Alpes, France

Ernesto Jiménez-Ruiz,
City, Univeristy of London, UK & SIRIUS, Univeristy of Oslo, Norway

Oktie Hassanzadeh,
IBM Research, USA

Cássia Trojahn,
IRIT, France

Program Committee

Alsayed Algergawy, Jena University, Germany
Manuel Atencia, University Grenoble Alpes & INRIA, France
Zohra Bellahsene, LIRMM, France
Jiaoyan Chen, University of Oxford, UK
Valerie Cross, Miami University, USA
Jérôme David, University Grenoble Alpes & INRIA, France
Gayo Diallo, University of Bordeaux, France
Warith Eddine Djeddi, LIPAH & LABGED, Tunisia
AnHai Doan, University of Wisconsin, USA
Alfio Ferrara, University of Milan, Italy
Marko Gulić, University of Rijeka, Croatia
Wei Hu, Nanjing University, China
Ryutaro Ichise, National Institute of Informatics, Japan
Antoine Isaac, Vrije Universiteit Amsterdam & Europeana, Netherlands
Marouen Kachroudi, Université de Tunis El Manar, Tunis
Simon Kocbek, University of Melbourne, Australia
Prodromos Kolyvakis, EPFL, Switzerland
Patrick Lambrix, Linköpings Universitet, Sweden
Oliver Lehmberg, University of Mannheim, Germany
Vincenzo Maltese, University of Trento, Italy
Fiona McNeill, University of Edinburgh, UK
Christian Meilicke, University of Mannheim, Germany

Peter Mork, MITRE, USA
Andriy Nikolov, Metaphacts GmbH, Germany
Axel Ngonga, University of Paderborn, Germany
George Papadakis, University of Athens, Greece
Catia Pesquita, University of Lisbon, Portugal
Henry Rosales-Méndez, University of Chile, Chile
Juan Sequeda, data.world, USA
Kavitha Srinivas, IBM, USA
Giorgos Stoilos, National Technical University of Athens, Greece
Pedro Szekely, University of Southern California, USA
Valentina Tamma, University of Liverpool, UK
Ludger van Elst, DFKI, Germany
Xingsi Xue, Fujian University of Technology, China
Ondřej Zamazal, Prague University of Economics, Czech Republic
Songmao Zhang, Chinese Academy of Sciences, China

Table of Contents

Long Technical Papers

Matching ontologies for air traffic management:
a comparison and reference alignment of the AIRM and NASA ATM ontologies
Audun Vennesland, Richard M. Keller, Christoph G. Schuetz,
Eduard Gringinger, Bernd Neumayr 1

Multi-view embedding for biomedical ontology matching
Weizhuo Li, Xuxiang Duan, Meng Wang, XiaoPing Zhang, Guilin Qi 13

Identifying mappings among knowledge graphs by formal concept analysis
Guowei Chen, Songmao Zhang 25

Short Technical Papers

Hypernym relation extraction for establishing subsumptions:
preliminary results on matching foundational ontologies
Mouna Kamel, Daniela Schmidt, Cássia Trojahn, Renata Vieira 36

Generating corrupted data sources for the evaluation of matching systems
Fiona McNeill, Diana Bental, Alasdair Gray,
Sabina Jedrzejczyk, Ahmad Alsadeeqi 41

OAEI Papers

Results of the Ontology Alignment Evaluation Initiative 2019
Alsayed Algergawy, Daniel Faria, Alfio Ferrara, Irini Fundulaki, Ian Harrow, Sven Hertling, Ernesto Jiménez-Ruiz, Naouel Karam, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Stefano Montanelli, Heiko Paulheim, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Elodie Thiéblin, Cássia Trojahn, Jana Vataščinová, Ondřej Zamazal, Lu Zhou 46

AnyGraphMatcher submission to the OAEI knowledge graph challenge 2019
Alexander Lütke 86

ALIN results for OAEI 2019
Jomar da Silva, Carla Delgado, Kate Revoredo, Fernanda Baião 94

AML and AMLC results for OAEI 2019
Daniel Faria, Catia Pesquita, Teemu Tervo, Francisco M. Couto, Isabel F. Cruz 101

AROA results for 2019 OAEI
Lu Zhou, Michelle Cheatham, Pascal Hitzler 107

CANARD complex matching system: results of the 2019 OAEI evaluation campaign
Elodie Thiéblin, Ollivier Haemmerlé, Cássia Trojahn 114

DOME results for OAEI 2019
Sven Hertling, Heiko Paulheim 123

EVOCROS: results for OAEI 2019
Juliana Medeiros Destro, Javier A. Vargas, Julio Cesar dos Reis, Ricardo da S. Torres 131

FCAMap-KG results for OAEI 2019
Fei Chang, Guowei Chen, Songmao Zhang 138

FTRLIM results for OAEI 2019
Xiaowen Wang, Yizhi Jiang, Yi Luo, Hongfei Fan, Hua Jiang, Hongming Zhu, Qin Liu 146

Lily results for OAEI 2019
Jiangheng Wu, Zhe Pan, Ce Zhang, Peng Wang 153

LogMap family participation in the OAEI 2019
Ernesto Jiménez-Ruiz 160

ONTMAT1: results for OAEI 2019
Saida Gherbi, Mohamed Tarek Khadir 164

POMap++ results for OAEI 2019:
fully automated machine learning approach for ontology matching
Amir Laadhar, Faiza Ghozzi, Imen Megdiche, Franck Ravat,
Olivier Teste, Faiez Gargouri 169

SANOM results for OAEI 2019
Majid Mohammadi, Amir Ahooye Atashin, Wout Hofman, Yao-Hua Tan 175

Wiktionary matcher
Jan Portisch, Michael Hladik, Heiko Paulheim 181

Posters

MultiKE: a multi-view knowledge graph embedding framework
for entity alignment
Wei Hu, Qingheng Zhang, Zequn Sun, Jiacheng Huang 189

MTab: matching tabular data to knowledge graph
with probability models
Phuc Nguyen, Natthawut Kertkeidkachorn,
Ryutaro Ichise, Hideaki Takeda 191

Generating referring expressions from knowledge graphs
Armita Khajeh Nassiri, Nathalie Pernelle, Fatiha Saïs 193

Semantic table interpretation using MantisTable
Marco Cremaschi, Anisa Rula, Alessandra Siano, Flavio De Paoli 195

Towards explainable entity matching via comparison queries
Alina Petrova, Egor V. Kostylev, Bernardo Cuenca Grau,
Ian Horrocks 197

Discovering expressive rules for
complex ontology matching and data interlinking
Manuel Atencia, Jérôme David, Jérôme Euzenat, Liliana Ibanescu,
Nathalie Pernelle, Fatiha Saïs, Elodie Thiéblin, Cássia Trojahn 199

Decentralized reasoning on a network of aligned ontologies
with link keys
Jérémy Lhez, Chan Le Duc, Thinh Dong, Myriam Lamolle 201

Matching Ontologies for Air Traffic Management: A Comparison and Reference Alignment of the AIRM and NASA ATM Ontologies

Audun Vennesland[1,2], Richard M. Keller[3], Christoph G. Schuetz[4(✉)], Eduard Gringinger[5], and Bernd Neumayr[4]

[1] Norwegian University of Science and Technology
audun.vennesland@ntnu.no
[2] SINTEF, Trondheim, Norway
[3] Intelligent Systems Division, NASA Ames Research Center, Moffett Field, CA, USA
rich.keller@nasa.gov
[4] Johannes Kepler University Linz, Linz, Austria
{christoph.schuetz,bernd.neumayr}@jku.at
[5] Frequentis AG, Vienna, Austria
eduard.gringinger@frequentis.com

Abstract. Air traffic management (ATM) relies on the timely exchange of information between stakeholders to ensure safety and efficiency of air traffic operations. In an effort to achieve semantic interoperability within ATM, the Single European Sky ATM Research (SESAR) program has developed the ATM Information Reference Model (AIRM), which individual information exchange models should comply with. An OWL representation of the AIRM – the AIRM Ontology (AIRM-O) – facilitates applications. Independently from the European efforts, the NASA Air Traffic Management Ontology (ATMONTO) has been developed as an RDF/OWL ontology representing ATM concepts to facilitate data integration and analysis in support of NASA aeronautics research. Conceptualization mismatches between the AIRM-O and ATMONTO ontologies – mostly due to different design decisions, but also as a consequence of the different regulatory systems and philosophies underlying ATM in Europe and the United States – pose a challenge to automatic ontology matching algorithms. In this paper, we describe mismatches between AIRM-O and ATMONTO, evaluate performance of automatic matching systems over these ontologies, and provide a manual reference alignment.

1 Introduction

Modern air traffic management (ATM) employs standardized models for the exchange of information required for seamless air traffic operations. Each exchange model has a different focus. The Aeronautical Information Exchange Model (AIXM) [1], for example, facilitates the representation of messages for pilots and air traffic controllers notifying of important events such as temporary runway closures and malfunctions of navigation aids. The exchange models are

subject to constant evolution in various standards working groups. In this regard, maintaining consistent co-evolution of the different exchange models is a necessity not only to guarantee efficiency of operations – by ensuring interoperability of systems – but also for safety reasons.

Recognizing the necessity of a common reference for the constantly evolving exchange models, the Single European Sky ATM Research (SESAR) program established the *ATM Information Reference Model* (AIRM) [25], developed under supervision of EUROCONTROL in an effort with industry and academia but meanwhile also adopted by the International Civil Aviation Organization (ICAO). The individual exchange models must ensure compliance with AIRM.

The *AIRM Ontology* (AIRM-O) [21] is an OWL ontology derived from the UML representation of AIRM in an effort to facilitate operationalization of AIRM. In this regard, previous work has investigated automatic compliance validation between exchange models and AIRM [22] as well as the annotation of ATM data sources with a semantic description of the contents [15].

The *NASA Air Traffic Management Ontology* (ATMONTO) [12, 13] supports NASA's aeronautics research activities by facilitating integration of data from various sources for analysis purposes. Developed independently from AIRM with a different purpose and under a different regulatory system – the United States instead of Europe – the question arises to what extent ATMONTO is actually compatible with AIRM-O.

In order to link AIRM-O and ATMONTO, we manually produced a reference alignment between these ontologies. In the course of the alignment process, we identified different types of mismatches between AIRM-O and ATMONTO, which we relate to existing mismatch classifications from literature. During the manual mapping process, we also experimented with state-of-the-art ontology matching systems. Some of the encountered mismatches pose a serious challenge for automatic ontology matching systems. According to the results from some of the benchmarks organised by the Ontology Alignment Evaluation Initiative (OAEI), the performance of ontology matching systems has improved significantly over recent years [7]. In some tracks, several of the competing systems achieve close to perfect F-measure [5], i.e., they are able to identify almost all relations in the track's ground truth alignment without producing false positives. Matching the two ATM ontologies, however, proved somewhat difficult for these systems. Some of the tested systems identified very few but correct relations whereas others identified a couple of more correct relations, but included too many incorrect relations. The reference alignment between ATMONTO and AIRM-O may serve the ontology matching community as a gold standard for improving and evaluating matching algorithms.

The remainder of this paper is organized as follows. In Sect. 2 we present relevant background information about the investigated ATM ontologies. In Sect. 3 we introduce a reference alignment between ATMONTO and AIRM-O. In Sect. 4 we identify mismatches between the ontologies. In Sect. 5 we evaluate performance of automatic matching systems. In Sect. 6 we review related work. We conclude with a summary and an outlook on future work.

2 Ontologies for Air Traffic Management

The AIRM addresses the issue of semantic interoperability between ATM systems through harmonized and agreed upon definitions of the information being exchanged in ATM [25]. The exchanged ATM information must comply with the AIRM definitions, the individual exchange models are aligned with the AIRM. AIRM is defined in UML, the various diagrams falling into the following subject fields: *AirTrafficOperations*, *Aircraft*, *AirspaceInfrastructure*, *BaseInfrastructure*, *Common*, *Environment*, *Flight*, *Meteorology*, *Stakeholders*, and *Surveillance*. The subject fields represent specific concerns of ATM.

In order to facilitate application of AIRM in practice, the SESAR exploratory research project BEST[6] developed the *AIRM Ontology* (AIRM-O) [21]. AIRM-O has been semi-automatically derived from the XML Metadata Interchange (XMI) representation of the AIRM UML diagrams using manual preprocessing and XSL Transformation (XSLT) scripts to obtain an OWL ontology. The transformation of the AIRM UML diagrams into an OWL ontology follows the Object Management Group's guidelines from the Ontology Definition Metamodel [17].

Independently from AIRM, ATMONTO was developed in the context of NASA's aeronautics research activities as a facilitator for data integration and analysis. ATMONTO supports semantic integration of ATM data being collected and analyzed at NASA for research and development purposes. The ontology functions as an integrative superstructure upon which to overlay data from multiple stove-piped aviation data sources, thus enabling cross-source queries that would be otherwise time-consuming and costly. ATMONTO includes a wide range of classes, properties, and relationships covering aspects of flight and navigation, aircraft equipment and systems, airspace infrastructure, meteorology, air traffic management initiatives, and other areas.

Development of ATMONTO followed a classic knowledge modeling approach. First, domain experts identified a core set of aviation data sources to be integrated. After an analysis of these sources, a proposed set of ATM concepts, properties, and relations was developed and presented to the experts for critique. The corresponding revisions led to an initial version of ATMONTO. Since this version was built in a bottom-up fashion driven by a need to accommodate the core data sources, the initial ontology did not represent the full complexity of the ATM domain. Gradually, additional data sources were incorporated, thereby revising and extending ATMONTO's set of concepts, properties, and relations. By the end of the development process, more than ten different data sources were covered by the ontology, and ATMONTO's structure had been generalized well beyond those sources. Although a general model of the ATM domain, ATMONTO's development was heavily driven by application requirements. In turn, AIRM-O's scope is overall broader than ATMONTO's since AIRM has been subject to a more coordinated standardization and governance process inside SESAR, harmonizing the various ATM information exchange models.

[6] Achieving the Benefits of SWIM by Making Smart Use of Semantic Technologies, https://project-best.eu/

3 Reference Alignment

In order to develop a reference alignment between AIRM-O and ATMONTO, a panel of six experts, each having experience within the ATM domain and knowledge of semantic technologies, collaboratively produced a mapping between concepts of the two ontologies. All the experts were asked to match each of the 157 classes in ATMONTO to corresponding classes in the larger AIRM-O – see Table 1 for statistics about the size of the ontologies – by making use of the experts' own domain knowledge as well as all available input, including descriptive class and property annotations in the ontologies and informative web resources such as Skybrary[7].

Table 1. Ontology Statistics

	Classes	Object Properties	Data Properties
ATMONTO	157	126	189
AIRM-O	915	1761	494

In addition to identifying equivalence classes, each expert also indicated subsumption relationships between concepts as well as potential mismatches of varying degree (see Sect. 4). After the initial matches were compiled, two of the five experts in the panel reviewed the matches for each ATMONTO class and produced a consensus mapping holding equivalence relations between classes from the ontologies. With the consensus mapping as a starting point, the reference alignment was developed using the following approach:

1. *Develop equivalence reference alignment.* The consensus mapping described above is formatted in RDF/XML according to the Alignment Format [3].
2. *Develop subsumption reference alignment.* Here, the same procedure as in the OAEI 2011 edition [4] was followed: The two source ontologies were merged into one single ontology in Protégé. Then OWL *equivalentClass* axioms consistent with the mapping described above were manually added between the corresponding classes in the merged ontology. An automated reasoner (HermiT) performed subsumption reasoning over the classes in the merged ontology in order to infer subsumption relations. In addition, subsumption mappings that were discovered in the manual mapping process but not identified by the reasoner were included in the reference alignment.
3. *Evaluate reference alignments.* Once both reference alignments were complete they were manually inspected for errors and inconsistencies.

The reference alignment between ATMONTO and AIRM-O [20] comes as two separate alignment files, one holding only equivalence relations and the other holding only subsumption relations. The equivalence reference alignment

[7] https://www.skybrary.aero/

contains 32 relations in total and the subsumption reference alignment contains 83 subsumption relations. Only direct subsumption relationships were considered in the subsumption reference alignment, following the convention used during the development of the reference alignment for the *Oriented Matching* track arranged in OAEI 2011 [4].

4 Mismatches between AIRM-O and ATMONTO

In the course of conducting the manual alignment of ATMONTO and AIRM-O (see Sect. 3), several of the identified candidate equivalence relations were considered "light matches" at first. In these cases, an equivalence relation between the classes was often deemed too strong – despite lexically similar class names hinting at a relation – given that the experts performed poorly on the alignment task – as judged by the two reviewing experts. Extensive discussions among the experts involved in the matching exercise revealed that similar class names were no guarantee of a correct match. In fact, in approximately 25% of the identified exact-match pairs in the final reference alignment, the class names *did not* have any words in common whereas in approximately 40% of the identified "light-match" candidate equivalence relations the class names *did* have words in common. This may explain partly why automated alignment techniques focusing on class name similarity did not perform particularly well (see Sect. 5).

The initially identified "light matches" between ATMONTO and AIRM-O actually represent *ontology mismatches*. Multiple classification systems for mismatchs with varying degrees of detail and often considerable overlap exist in literature. Figure 1 shows a classification of mismatch types synthesized from Klein [14] and Visser et al. [23, 24] along with mismatch types encountered during the manual matching between ATMONTO and AIRM-O. Notwithstanding the differences between classification systems, there seems to be consensus that the development of an ontology involves two separate processes and, correspondingly, two broad categories of mismatches can be distinguished [23, 24]. First, *conceptualization mismatches* are the result of different interpretations of the represented domain, leading to different classes, individuals, and relations being modeled in different ontologies for the same domain. *Explication mismatches*, on the other hand, are the result of different specifications of domain interpretations in form of different terms, modeling styles, and encodings being employed.

One category of conceptualization mismatches concerns differences in *model coverage and scope* between ontologies from the same domain, which occur when two ontologies cover different parts of that domain or the same part at different levels of detail. In this regard, a *structure mismatch* occurs when two ontologies distinguishing the same set of classes differ in how they are structured through relations; we could not find a clear case of structure mismatch between ATMONTO and AIRM-O. A mismatch concerning *differing levels of detail* occurs when one class is modeled in more depth and with greater fidelity than the other. The $ASPMeteorologicalCondition$ class from ATMONTO and $AerodromeCondition$ from AIRM-O, for example, both represent meteorological

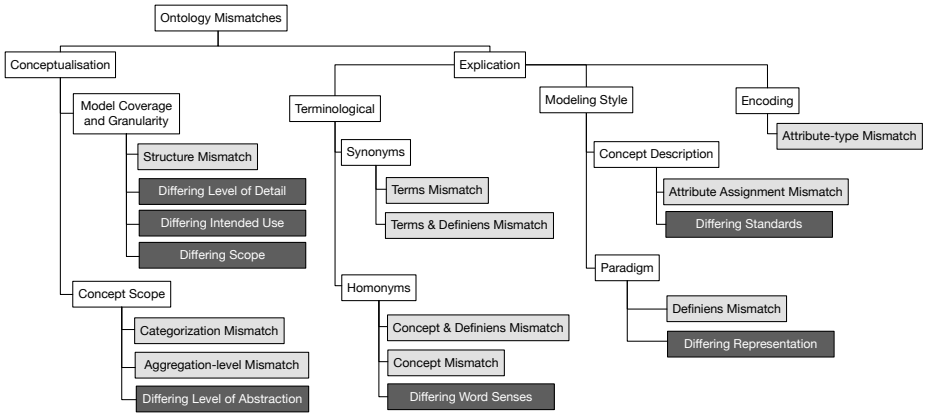


Fig. 1. Classification of ontology mismatches, synthesized from Klein [14] (white) and Visser et al. [23, 24] (light grey), extended with mismatch types encountered when mapping ATMONTO to AIRM-O (dark grey).

information. $ASPMeteorologicalCondition$, however, is more detailed, comprising all aspects of sky, wind, visibility, and weather whereas $AerodromeConditon$ is limited to sky conditions. Different properties and relations of similar classes may also reflect differences in how the classes are to be used in the context of a domain application (*differing intended uses*). For example, $ReRouteSegment$ in ATMONTO describes an alternative air route option for contingency planning purposes, whereas $RouteSegment$ describes an actual portion of a route being flown. Eventually, the *differing scope* of ontologies may result in a class from the source ontology lacking a matching class in the target ontology because the class from the source ontology lies outside the defined scope of the target ontology. An example of a differing scope is the missing equivalent in AIRM-O for the class $DelayModel$ in ATMONTO, which specifies a numerical model of airspace delay under specific traffic conditions. There is no matching class in AIRM-O because modeling concerns fall outside the scope of this ontology.

A *concept scope* conceptualization mismatch occurs when two classes seem to represent the same concept, yet do not cover exactly the same instances, although the classes intersect. Categorization mismatches and aggregation-level mismatches fall into the concept scope mismatch category. A categorization mismatch occurs when two ontologies include the same class, but each ontology decomposes the class into different subclasses. ATMONTO's $Airport$ is equivalent to AIRM-O's $Aerodrome$, however due to different geographical and application-wise scope $Airport$ includes the subclasses $USairport$ and $InternationalAirport$ whereas $Aerodrome$ has no such subclasses. An aggregation-level mismatch occurs when two ontologies define the same underlying concept using classes at different levels of abstraction. A *differing level of abstraction* is encountered when the matched classes intersect but some instances are outside the intersection. Consider, for example, $AviationIndustryManufacturer$ in ATMONTO and

AerospaceManufacturer in AIRM-O. In this case, the term "Aerospace" has a broader meaning than "Aviation", hinting at a subsumption relation.

The class of explication mismatches encompasses *terminological*, *modeling style*, and *encoding* mismatches. In this regard, an encoding mismatch relates to how the ontologies employ different formatting when describing instances, e.g., describing an instance either in miles or kilometres [14]; we omit this mismatch type in the remainder of this analysis. More relevant for our analysis are the terminological and modeling-style mismatches identified by Visser et al. [23, 24], which occur due to different knowledge definitions used in the ontologies and their associated concepts.

The category of terminological mismatches comprises mismatches related to *synonyms* and *homonyms*. The synonym mismatch as explained by Klein [14] refers to two lexically different terms in fact meaning the same thing (e.g. 'Airport/Heliport' versus 'Aerodrome'), so we do not consider this a real mismatch in our analysis. Term mismatches as well as terms-and-definiens mismatches defined by Visser et al. [23, 24] belong to the synonym mismatches. A term mismatch occurs when the definitions share the same concept and the same definiens, but the terms are different. Correspondingly, a term-and-definiens mismatch occurs when the definitions refer to the same underlying concept, but the terms and definiens are different. The relation between *Airport* in ATMONTO and *Aerodrome* in AIRM-O could also be considered a terms-and-definiens mismatch.

Mismatches related to homonyms occur when the meaning of two identical terms is different (e.g. the term 'Conductor' has a different meaning in music than in electrical engineering). We refer to homonym mismatches proper as *differing word senses*. There were a few incidents of homonymy that complicated the alignment process for ATMONTO and AIRM-O. For example, the term "Flow" had a slightly different meaning in ATMONTO and AIRM-O. In AIRM-O, a flow is a traffic pattern, while in ATMONTO flow is a concrete measurement of the number of aircraft per time unit traversing a volume of airspace.The classes have an exact or close lexical match, but the two classes correspond to two different word senses.

Modeling style mismatches are further decomposed into *concept description* and *paradigm* mismatches. A concept description mismatch occurs when two similar concepts are modelled differently, e.g., that the same intention is modelled through the use of properties in one ontology and by using distinct sub-classes for the same target values in the other ontology [6]. A specific type of concept description mismatch between ATMONTO and AIRM-O is classes with similar names defining different versions of the same concept based on differing technical standards adopted by ontology developers, e.g., by FAA and EUROCONTROL. Finally, paradigm mismatches refer to how different paradigms can be used to represent concepts such as time, action, plans, causality, propositional attitudes, etc. For example, one ontology might use temporal representations based on interval logic, while another might use a representation based on points [6]. Paradigm mismatches relate to what we call "differing representation", and one example of such a mismatch is between *PlannedFlightRoute* in ATMONTO

and *Trajectory* in AIRM-O. These two classes are used to represent the planned aircraft trajectory (or flight plan). In AIRM-O, the planned trajectory is composed of a sequence of trajectory points, elements, segments, and constraints. In ATMONTO, the flight plan is specified using a hierarchically decomposable route structure. These are fundamentally different methods of representing a planned route, based on different conceptual models of what constitutes a route.

5 Performance of Automatic Matching Systems

We challenged three matching systems that normally rank highly on several tracks of the OAEI campaigns on the equivalence reference alignment:

- AgreementMakerLight (AML) [9]. We ran AML using the GUI version from 2016[8] and the "Automatic Match" mode, letting AML handle the configuration of individual matching algorithms and external background sources (e.g. WordNet). AML includes terminological, structural and lexical matchers and uses WordNet as a general-purpose lexical resource as well as the Doid and Uberon ontologies for matching of biomedical ontologies. Property relations included in the produced alignment were disregarded when evaluating the performance of AML.
- LogMap [11]. We used the latest available standalone distribution of LogMap[9] with default matching parameters. LogMap combines terminological matching with capabilities for diagnosing and repairing incoherent alignments. Optionally, LogMap can also employ external resources such as WordNet. As with AML there were some property relations included in the produced alignment, which we do not consider in the evaluation.
- YAM++ [16]. YAM++ is provided as a web application[10]. We used the default matcher parameters, which included both an element-level and a structure-level matching algorithm.

The evaluation results from running the matching systems on the equivalence reference alignment are shown in Figure 2. As the figure shows, all three systems manage to avoid many false positives, especially LogMap which obtains perfect precision with no false positives. All three systems obtain a recall of 0.31. The results reveal that all three matching systems are able to correctly detect the true positive relations where the source and target classes are exact string matches. All three matchers also capture one relation where the source class (*SID*) is an acronym of the target class (*StandardInstrumentDeparture*) due to the fact that "Standard Instrument Departure" is expressed in the label of the source class. The remaining relations in the reference alignment are not detected by these systems.

A closer inspection of the alignments produced by these three matching systems with respect to the equivalence reference alignment reveals that the following factors contribute to making this a challenging dataset:

[8] There was an issue with the dependency to the Gephi Toolkit that prevented us from using the most recent version of AML.
[9] https://sourceforge.net/projects/logmap-matcher/files/
[10] http://yamplusplus.lirmm.fr/index

– *Domain-specific and technical terminology.* Most of the classes in both ontologies describe aviation-specific concepts and technical terms. Often the class names and their natural language definitions include acronyms and abbreviations used only in aviation. Considering that typically used lexical resources (such as the aforementioned WordNet) have low coverage of technical terminology, this constitutes a challenge for matching systems.

– *Compound class names.* Several of the classes involved in the relations represented in the reference alignment contains equal substrings, a feature often exploited by string-matching techniques. However, in most relations one or both class names are compound words, such as *PhysicalRunway* - *Runway* or *AircraftModel* - *AircraftMakeModelSeries*, resulting in a low similarity scores for algorithms based on basic substring analysis. Here, a more comprehensive string-based analysis is required to identify such relations, possibly resulting in the unwanted effect that additional false positive relations are being included in the computed alignment as well.

– *Synonymy, homonymy and polysemy.* The two ontologies use synonymous terms for concepts with the same meaning (e.g. Airport vs. Aerodrome). Synonymy can often be resolved using lexicons or other external sources (e.g. other ontologies). Homonymy and polysemy are more of a challenge to solve. Some of the class names in these two ontologies can have a different meaning outside the ATM domain. Examples of this are Gate, Taxi or Star (which is short for *Standard Terminal Arrival Route* in the ATM world) and such challenges are not addressed through the use of lexicons such as WordNet.

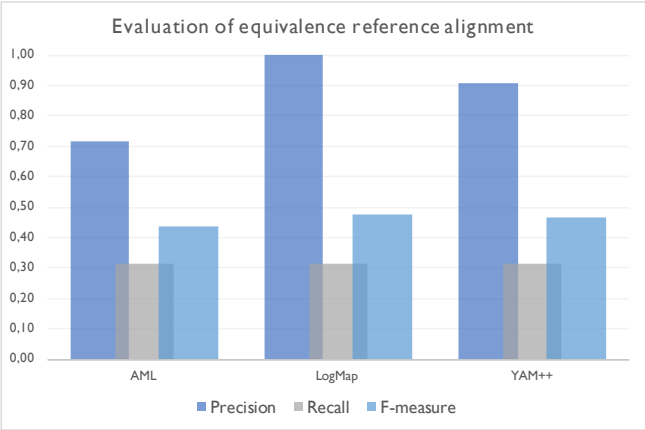


Fig. 2. Performance of selected state-of-the-art matchers over ATMONTO and AIRM-O

6 Related Work

Evaluation datasets that include reference alignments declaring the correct set of mappings between ontologies are important for the continued improvement of ontology matching techniques. The OAEI provides an annual standardised evaluation process for matching system. However, with only a few exceptions over the years, the OAEI tracks mainly involve one-to-one equivalence relations, neglecting other semantic relations and complex correspondences whose identification is important for more profound integration processes [8, 18]. One of these OAEI tracks is the Conference Track, a widely used benchmark for ontology matching systems, that since its inception in 2005 has been subject to many revisions [26]. This track now includes 16 ontologies describing conference organization and there are two versions of reference alignments, all holding one-to-one equivalence relations. The first version is referred to as "crisp" alignments where all confidence values are 1.0. The second version is referred to as an "uncertain" version of the reference alignment where the confidence values reflect the opinion from a group of human experts [7].

For the 2018 OAEI campaign, a complex alignment track was launched, offering reference alignments holding complex relations in four different datasets. One of the datasets included complex reference alignments for some of the ontologies in the Conference Track [19]. The other datasets represented real-world ontologies from the domains of hydrography, plants and species, and geoscience. Having real-world ontologies in benchmarks is important because such ontologies may expose issues arising in practice which may be overlooked by the developers of (semi-)artificial benchmarks [27].

7 Summary and Future Work

We contrasted AIRM-O with the ATMONTO. Mismatches between these ontologies coupled with the complex and diverse nature of the ATM domain, which covers many technical subject fields, renders automatic ontology matching difficult. The presented manual alignment of AIRM-O and ATMONTO potentially facilitates integration of datasets in different formats, e.g., NASA aeronautics research data with ATM information in the operational System Wide Information Management (SWIM) network. As a byproduct, the ontology matching community gains access to a reference alignment for two complex real-world ontologies from the ATM domain. We refer to a separate publication [10] for a more detailed comparison of AIRM-O and ATMONTO from an ATM perspective.

Future work will investigate the potential for complex reference alignments between AIRM-O and ATMONTO beyond simple equivalence and subsumption relations. using the Expressive and Declarative Ontology Alignment Language (EDOAL) [2]. During the manual mapping process, we identified a large number of complex relations, e.g., class-to-property relations and many-to-many relations, which additional reference alignments can be developed from. In this regard, complex matching represents an area with a potential for significantly advancing the state-of-the-art in ontology matching.

Acknowledgments. We thank Scott Wilson from EUROCONTROL and Joe Gorman from SINTEF for their contributions to the reference alignment. Part of this work was conducted as part of the BEST project. This project received funding from the SESAR Joint Undertaking under grant agreement No 699298 under the European Union's Horizon 2020 research and innovation program. This work was also supported by the NASA Airspace Operations and Safety Program. The views expressed in this paper are those of the authors.

References

1. Aeronautical Information Exchange Model. http://aixm.aero/ (Accessed: 26 August 2019)
2. EDOAL: Expressive and Declarative Ontology Alignment Language. http://alignapi.gforge.inria.fr/edoal.html (Accessed: 26 August 2019)
3. A format for ontology alignment. http://alignapi.gforge.inria.fr/format.html (Accessed: 26 August 2019)
4. Ontology Alignment Evaluation Initiative: Oriented Matching 2011. http://oaei.ontologymatching.org/2011/oriented/index.html (Accessed: 26 August 2019)
5. Achichi, M., et al.: Results of the Ontology Alignment Evaluation Initiative 2017. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 12th International Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 2032, pp. 61–113. CEUR-WS.org (2017)
6. Chalupsky, H.: OntoMorph: A translation system for symbolic knowledge. In: Cohn, A.G., Giunchiglia, F., Selman, B. (eds.) Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2000). pp. 471–482. Morgan Kaufmann
7. Cheatham, M., Hitzler, P.: Conference v2.0: An uncertain version of the OAEI conference benchmark. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C.A., Vrandecic, D., Groth, P.T., Noy, N.F., Janowicz, K., Goble, C.A. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 33–48. Springer (2014)
8. Cruz, I.F., Palmonari, M., Caimi, F., Stroe, C.: Building linked ontologies with high precision using subclass mapping discovery. Artificial Intelligence Review **40**(2), 127–145 (2013)
9. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 527–541. Springer (2013)
10. Gringinger, E., Keller, R.M., Vennesland, A., Schuetz, C.G., Neumayr, B.: A comparative study of two complex ontologies in air traffic management. In: Proceedings of the AIAA/IEEE 38th Digital Avionics Systems Conference (DASC) (2019), in press
11. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and scalable ontology matching. In: ISWC 2011. LNCS, vol. 7031, pp. 273–288. Springer (2011)

12. Keller, R.M.: The NASA Air Traffic Management Ontology. https://data.nasa.gov/ontologies/atmonto (March 2018)
13. Keller, R.M.: Building a knowledge graph for the air traffic management community. In: Companion of The 2019 World Wide Web Conference. pp. 700–704 (2019)
14. Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing (2001)
15. Neumayr, B., Gringinger, E., Schuetz, C.G., Schrefl, M., Wilson, S., Vennesland, A.: Semantic data containers for realizing the full potential of system wide information management. In: Proceedings of the 36th IEEE/AIAA Digital Avionics Systems Conference (DASC) (2017)
16. Ngo, D., Bellahsene, Z.: Overview of YAM++ – (not) yet another matcher for ontology alignment task. Web Semantics: Science, Services and Agents on the World Wide Web **41**, 30–49 (2016)
17. Object Management Group: Ontology Definition Metamodel v1.1 (2014), https://www.omg.org/spec/ODM/1.1/
18. Spiliopoulos, V., Vouros, G.A., Karkaletsis, V.: On the discovery of subsumption relations for the alignment of ontologies. Journal of Web Semantics **8**(1), 69–88 (2010)
19. Thiéblin, É., Haemmerlé, O., Hernandez, N., Trojahn, C.: Task-oriented complex ontology alignment: Two alignment evaluation sets. In: European Semantic Web Conference. pp. 655–670. Springer (2018)
20. Vennesland, A., Keller, R., Gringinger, E., Schuetz, C.G., Neumayr, B., Wilson, S., Gorman, J.: ATMONTO2AIRM A reference alignment between the NASA ATM Ontolgy and the ATM Information Reference Model Ontology. https://w3id.org/airm-o/atmonto2airm/ (2019)
21. Vennesland, A., Neumayr, B., Schuetz, C.G., Savulov, A., Wilson, S., Gringinger, E., Gorman, J.: AIRM-O – ATM Information Reference Model Ontology. https://w3id.org/airm-o/ontology/ (2017)
22. Vennesland, A., Gorman, J., Wilson, S., Neumayr, B., Schuetz, C.G.: Automated compliance verification in ATM using principles from ontology matching. In: Aveiro, D., Dietz, J.L.G., Filipe, J. (eds.) Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K. pp. 39–50. SciTePress (2018)
23. Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M., Shave, M.J.R.: An analysis of ontology mismatches; heterogeneity versus interoperability. In: Proceedings of the AAAI 1997 Spring Symposium on Ontological Engineering. pp. 164–72 (1997)
24. Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M., Shave, M.J.R.: Assessing heterogeneity by classifying ontology mismatches. In: Guarino, N. (ed.) Proceedings of the 1st International Conference on Formal Ontology in Information Systems. IOS Press (1998)
25. Wilson, S., Suzi, R., Van der Stricht, S.: The SESAR ATM information reference model within the new ATM system. In: Proceedings of the 2014 Integrated Communications, Navigation and Surveillance Conference (ICNS) Conference. pp. L3–1–L3–13 (2014)
26. Zamazal, O., Svátek, V.: The ten-year ontofarm and its fertilization within the onto-sphere. Journal of Web Semantics **43**, 46–53 (2017)
27. Zhou, L., Cheatham, M., Krisnadhi, A., Hitzler, P.: A complex alignment benchmark: GeoLink dataset. In: Vrandecic, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L., Simperl, E. (eds.) ISWC 2018. LNCS, vol. 11137, pp. 273–288. Springer (2018)

Multi-view Embedding for Biomedical Ontology Matching*

Weizhuo Li[1,2], Xuxiang Duan[3], Meng Wang[1,2], XiaoPing Zhang[4(✉)], and Guilin Qi[1,2]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China.
liweizhuo@amss.ac.cn, {meng.wang,gqi}@seu.edu.cn
[2] Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China.
[3] School of Mathematical Sciences, Chongqing Normal University, Chongqing, China.
duanxx9156@163.com
[4] China Academy of Chinese Medical Sciences, Beijing, China.
xiao_ping_zhang@139.com

Abstract. The goal of ontology matching (OM) is to identify mappings between entities from different yet overlapping ontologies so as to facilitate semantic integration, reuse and interoperability. Representation learning methods have been applied to OM tasks with the development of deep learning. However, there still exist two limitations. Firstly, these methods are of poor capability of encoding sparse entities in ontologies. Secondly, most methods focus on the terminological-based features to learn word vectors for discovering mappings, but they do not make full use of structural relations in ontologies. It may cause that these methods heavily rely on the performance of pre-training and are limited without dictionaries or sufficient textual corpora. To address these issues, we propose an alternative ontology matching framework called MultiOM, which models the matching process by embedding techniques from multiple views. We design different loss functions based on cross-entropy to learn the vector representations of concepts, and further propose a novel negative sampling skill tailored for the structural relations asserted in ontologies. The preliminary result on real-world biomedical ontologies indicates that MultiOM is competitive with several OAEI top-ranked systems in terms of F1-measure.

Key words: Ontology Matching, Embedding, Cross-Entropy, Negative Sampling

1 Introduction

In the Semantic Web, ontologies aim to model domain conceptualizations so that applications built upon them can be compatible with each other by sharing the same meanings. Life science is one of the most prominent application areas of ontology technology. Many biomedical ontologies have been developed and utilized in real-world systems including Foundational Model of Anatomy (FMA)[5], Adult Mouse Anatomy (MA)

 This work was partially supported by the National Key Research and Development Program of China under grant (2018YFC0830200) the Natural Science Foundation of China grants (U1736204), the Fundamental Research Funds for the Central public welfare research institutes (ZZ11-064), the Fundamental Research Funds for the Central Universities (3209009601).

[5] http://si.washington.edu/projects/fma

for anatomy[6], National Cancer Institute Thesaurus (NCI)[7] for disease and Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT)[8] for clinical medicine. To integrate and migrate data among applications, it is crucial to first establish mappings between the entities of their respective ontologies. As ontologies in the same domain are often developed for various purposes, there exist several differences in coverage, granularity, naming, structure and many other aspects. It severely impedes the sharing and reuse of ontologies. Therefore, ontology matching (OM) techniques devote to identify mappings across ontologies in order to alleviate above heterogeneities [1].

In the last ten years, many automatic systems are developed so as to discover mappings between independently developed ontologies and obtain encouraging results (see [2, 3] for a comprehensive and up-to-date survey). Up to now, the mainstream methods (e.g., LogMap [4], AML [5], FCA-Map [6], XMap [7]) still focus on engineering features from terminological, structural, extensional (individuals of concepts) information and external resource [1]. These features are utilized to compute the similarities of ontological entities (i.e., concepts, properties, individuals) for guiding the ontology matching. With the development of deep learning [8], there also exist several works (e.g., ERSOM [9], DeepAlignment [10], SCBOW + DAE(O) [11] OntoEmma [12]) that try to shift from feature engineering to representation learning. The assumption is that semantically similar or related words appear in similar contexts. Therefore, word vectors own the potentials that can bring significant value to OM given the fact that a great deal of ontological information comes in textual form [10]. Nevertheless, there still exist two challenges that need to be solved:

- **Sparsity Problem for Embedding Learning**: One of the main difficulties for embedding learning is of poor capability of encoding sparse entities. Even in large-scale medical ontologies with lots of relations, most knowledge graph embedding techniques (e.g., TransE [13]) are still not applicable. Zhang et al. [14] observed that the prediction results of entities were highly related to their frequency, and the results of sparse entities were much worse than those of frequent ones.
- **Limitation Problem for External Resource**: Thesaurus is one kind of external resource that is usually employed in matching systems such as WordNet [15], UMLS Metathesaurus[9]. In addition, textual descriptions can also be employed for ontology matching [11, 12]. Nevertheless, these methods based on representation learning rely heavily on the performance of pre-training. Therefore, it may limit their scalability if there exist no dictionaries or sufficient textual corpora.

To address above problems, we propose MultiOM, an alternative ontology matching framework based on embedding techniques from multiple views. The underlying idea is to divide the process of OM into different modules (i.e., lexical-based module, structural-based module, resource-based module) and employ embedding techniques to soften these modules. Existing works [16, 17] show that identifying multiple views can sufficiently represent the data and improve the accuracy and robustness of corresponding tasks. Therefore, we employ this idea to characterize the process of OM and try to alleviate the sparsity problem for embedding learning indirectly. More precisely, different loss functions are designed based on cross-entropy to model different views

[6] http://informatics.jax.org/vocab/gxd/ma_ontology
[7] https://ncit.nci.nih.gov/
[8] http://www.snomed.org/snomed-ct/
[9] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html

among ontologies and learn the vector representations of ontological entities. With continuous vector representation, we can obtain more similar concepts and discover more potential mappings among ontologies. Furthermore, we design a novel negative sampling tailored for structural relations (e.g., *subclassOf* relations, *disjointWith* relations) asserted in ontologies, which can obtain better vector representations of entities for OM.

The contributions of our study are summarized as follows.

- We propose an alternative ontology matching framework with embedding techniques from multiple views, and design loss functions based on cross-entropy to model different views for learning vector representations of ontological entities.
- We design a novel negative sampling skill tailored for structural relations asserted in ontologies, which can obtain better vector representations of concepts.
- We implement our method and evaluate it on real-world biomedical ontologies. The preliminary result indicates that MultiOM is competitive with several OAEI top-ranked systems in terms of F1-measure.

2 Related work

2.1 Feature-based methods for biomedical ontology matching

There exist various feature-based strategies applied on the scenarios biomedical ontology matching, including terminological-based features, structural-based features and employing external semantic thesauruses for discovering semantically similar entities.

LogMap [4] relies on lexical and structural indexes to enhance its scalability. To scale to large ontologies and minimize the number of logical errors in the aligned ontologies, LogMap uses a horn propositional logic representation of the extended hierarchy of each ontology together with all existing mappings and employs Dowling-Gallier algorithm to model propositional horn satisfiability.

AML [5] is originally developed to tackle the challenges of matching biomedical ontologies. It employs various sophisticated features and domain-specific thesauruses to perform OM. Besides, AML introduces a modularization-based technique to extract the core fragments of the ontologies that contain solely the necessary classes and relations caused by disjoint restrictions, which can repair the incoherent alignments effectively.

FCA-Map [6] is an ontology matching system based on formal concept analysis (FCA), in which five types of formal contexts are constructed in an incremental way, and their derived concept lattices are used to cluster the commonalities among classes and properties at various lexical and structural levels, respectively.

XMap [7] is a scalable matching system that implements parallel processing techniques to enable the composition of basic sophisticated features. It also relies on the employment of external resources such as UMLS Metathesarus to improve the performance of ontology matching.

PhenomeNet [18] exploits an axiom-based approach for aligning phenotype ontologies, which makes use of the PATO ontology and Entity-Quality definition patterns so as to complement several shortcomings of feature-based methods.

Feature-based methods mainly employ crafting features of the data to achieve specific tasks. Unfortunately, these hand-crafted features will be limited for a given task and face the bottleneck of improvement. Cheatham and Hitzler showed that the performance of ontology matching based on such engineered features varies greatly with the domain described by ontologies [19]. As a complement to feature engineering, continuous vectors representing ontological entities can capture the potential associations among features, which is helpful to discover more mappings among ontologies.

2.2 Representation learning methods for biomedical ontology matching

Representation learning have so far limited impacts on OM, specifically in biomedical ontologies. To the best of our knowledge, only five approaches have explored the use of unsupervised representation learning techniques for ontology matching.

Zhang et al. [20] is one of the first that investigate the use of word vectors for ontology matching. They align ontologies based on word2vec vectors [21] trained on Wikipedia. In addition, they use the semantic transformations to complement the lexical information such as names, labels, comments and describing entities. The strategy of entity matching is based on maximum similarity.

Xiang et al. [9] propose an entity representation learning algorithm based on Stacked Auto-Encoders, called ERSOM. To describe an ontological entity (i.e., concept, property), They design a combination of its ID, labels, comments, structural relations and related individuals. The similarity of entities is computed with a fixed point algorithm. Finally, ERSOM generates an alignment based on the stable marriage strategy.

DeepAlignment [10] is an unsupervised matching system, which refines pre-trained word vectors aiming at deriving the descriptions of entities for OM. To represent the ontological entities better, the authors represent words by learning their representations and using synonymy and antonymy constraints extracted from general lexical resources and information captured implicitly in ontologies.

SCBOW + DAE(O) [11] is representation learning framework based on terminological embeddings, in which the retrofitted word vectors are introduced and learned by the domain knowledge encoded in ontologies and semantic lexicons. In addition, SCBOW + DAE(O) incorporates an outlier detection mechanism based on a denoising autoencoder that is shown to improve the performance of alignments.

Wang et al. [12] propose a neural architecture tailored for biomedical ontology matching called OntoEmma, It can encode a variety of information and derive large amounts of labeled data for training the model. Moreover, they utilize natural language texts associated with entities to further improve the quality of alignments.

However, there exist two limitations for above methods. One is the sparsity problem of structural relations. To avoid the poor capability of encoding sparse relations, above methods prefer terminological-based features to learn word vectors for discovering mappings, but they do not make full use of structural relations in ontologies. The other is that these methods rely heavily on the performance of pre-training, which may limit their scalability if there exist no dictionaries or sufficient textual corpora.

3 Muti-view Embedding for Biomedical Ontology Matching

In the scenario of biomedical ontology matching, matching systems mainly focus on mappings of concepts with equivalent relations $(C_i, C_j, \equiv, n)$. Thus, in the remainder of the paper, we only consider these type of mapping for biomedical ontology matching.

3.1 MultiOM

Existing works [16, 17] show that identifying multiple views that can sufficiently represent the data and improve the accuracy and robustness of corresponding tasks. Inspired by their works, we characterize the process of OM from multiple views and try to alleviate the sparsity problem for embedding learning indirectly.

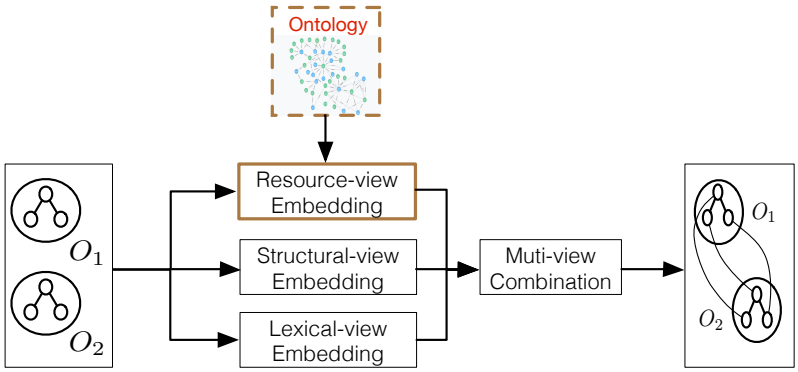


Fig. 1: The framework of MultiOM

The framework of MutiOM is shown in Fig. 1. Given two biomedical ontologies O_1 and O_2, we first extract the concepts and their information of ontologies. Then, we divide the process of OM into three embedding modules from different views, which compose of lexical-view embedding, structural-view embedding and resource-view embedding. Domain ontologies in the resource-based module, are treated as bridges to connect source ontology and target one for discovering more potential mappings. With a designed combination strategy based on mutual assessment, we obtain a final alignment among given ontologies.

Different from feature-based methods, we utilize ontological information to learn the continuous vector representations of concepts by embedding techniques, which can discover more potential mappings among ontologies. There exist different granularity of vector representations of modules in MultiOM. In lexical-based module, each concept is divided into several tokens $\{t_1, t_2, ..., t_n\}$ that are represented as k-dimensional continuous vectors $t_i, t_j \in \mathbb{R}^k$. The similarity of concepts is measured based on these word vectors by the designed algorithm. Relatively, for structural-based module and resource-based module, each concept C is represented as a d-dimensional continuous vector $\mathbf{C} \in \mathbb{R}^d$, and their similarities are calculated based on cosine measure.

Lexical-view Embedding The lexical-view embedding module is mainly based on TF-IDF algorithm, which is one of the most effective string similarity metrics for ontology matching [19]. According to the assumption of TF-IDF, concepts in one ontology can be represented as a bag of tokens. Then, every concept C_i is regarded as a document and the tokens $\{t_1, t_2, ..., t_l\}$ of each concept are treated as terms. Inspired by the idea soft TF-IDF [19], we propose an embedding-based TF-IDF strategy to calculate the similarities of concepts, More precisely, the similarity of each concept pair is calculated according to the similarities of their tokens, which is obtained based on the cosine measure of tokens' vectors representations rather than the string equivalent of them. The corresponding formula is defined as follows.

$$Sim(C_1, C_2) = \sum_{i=1} w_i \cdot \arg\max_j cos(\mathbf{t_{1i}}, \mathbf{t_{2j}}), \tag{1}$$

where C_1 and C_2 are concepts from O_1 and O_2, $\mathbf{t_{1i}}$ and $\mathbf{t_{2j}}$ are vector representations of tokens t_{1i} and t_{2i} that belong to C_1 and C_2. w_i is a weight of token $\mathbf{t_{1i}}$ in C_1 that is

calculated as follows.

$$w_i = \frac{\text{TFIDF}(t_{1i})}{\sum_{l=1}^{n} \text{TFIDF}(t_{1l})}, \tag{2}$$

where n is the number of tokens, TFIDF$(\cdot)$ returns the TF-IDF value of each token.

As cosine measure of $\mathbf{t_{1i}}$ and $\mathbf{t_{2j}}$ is a continuous value, so this embedding-based TF-IDF strategy is able to obtain more similar concepts and discover more potential mappings. Nevertheless, our softened strategy depends on the quality of embedding of tokens and may generate more wrong mappings. Therefore, we utilize pre-training vectors to cover the tokens of ontologies as soon as possible (see Section 4.2). On the other hand, we employ the mappings generated by other embedding modules to assess the quality of these mappings in lexical-view module (see Section 3.3).

Structural-view Embedding As mentioned before, most proposed methods focus on the terminological-based features to learn word vectors for ontology matching, but they do not make full use of structural relations in ontologies. Relatively, we try to generate mappings from the structural view. To obtain more candidate mappings for training embedding of concepts, we assume that the mappings generated by equivalent strings or their synonym labels are correct, and define a loss function based on cross-entropy to optimize the vector representations of concepts. The loss function is defined as follows.

$$l_{SE} = - \sum_{(C_1, C_2, \equiv, 1.0) \in \mathcal{M}} log f_{SE}(C_1, C_2) - \sum_{(C_1', C_2', \equiv, 1.0) \in \mathcal{M}'} log(1 - f_{SE}(C_1', C_2')), \tag{3}$$

where $\mathcal{M}$ is a set of candidate mappings $\{(C_1, C_2, \equiv, 1.0)\}$ generated by our assumption, $\mathcal{M}'$ is a set of negative mappings. We employ the negative sampling skill [13] to generate $\mathcal{M}'$ for training the loss function. For each mapping $(C_i, C_j, \equiv, 1.0) \in \mathcal{M}$, we corrupt it and randomly replace C_i or C_j to generate a negative triple $(C_i{}', C_j, \equiv, 1.0)$ or $(C_i, C_j{}', \equiv, 1.0)$. $f_{SE}(C_1, C_2)$ is a score function defined in Eq. 4 to calculate the score of concept pairs, where $\mathbf{C_1}, \mathbf{C_2} \in \mathbb{R}^d$ are d-dimensional continuous vectors of concepts C_1 and C_2 from different ontologies, $|| \cdot ||_2$ is the L_2-norm. We hope that $f_{SE}(C_1, C_2)$ is large if concepts C_1 and C_2 are similar.

$$f_{SE}(C_1, C_2) = 2 \cdot \frac{1}{1 + e^{(||\mathbf{C_1} - \mathbf{C_2}||_2)}}. \tag{4}$$

Furthermore, we design a negative sampling skill tailored for structural relations asserted in ontologies (e.g., *subclassOf* relations, *disjointWith*) relations. Unlike the uniform negative sampling method that samples its replacer from all the concepts, we limit the sampling scope to a group of candidates. More precisely, for each mapping $(C_i, C_j, \equiv, 1.0) \in \mathcal{M}$, if there exist *subclassOf* relations (e.g., $(C_i{}'$, *subclassOf*, $C_i)$ or $(C_j{}'$, *subclassOf*, $C_j)$) asserted in ontologies, we need to exclude this replace case. Relatively, for *disjointWith* relations (e.g.,$(C_i{}'$, *disjointWith*, $C_i)$ or $(C_j$, *disjointWith*, $C_j{}')$), we need to give the highest priority to these relations for replace cases (see Section 4.2). With these constrains for negative sampling, we can obtain better vector representations of concepts for ontology matching.

Resource-view Embedding Inspired by the work in [22], we consider external ontology as a bridge to connect two concepts from source ontology and target one. We

observe that there exist many different yet overlapping biomedical ontologies such as MA—NCI—FMA, FMA—NCI—SNOMED-CT. Compared with textual descriptions or thesaurus, ontologies as external resources can provide some structural assertions, which is helpful to refine the quality of discovered mappings [22]. Nevertheless, the original idea is mainly based on string equality, which may not discover more similar concepts. Therefore, we employ embedding techniques to soft this framework to discover more potential mappings from this view.

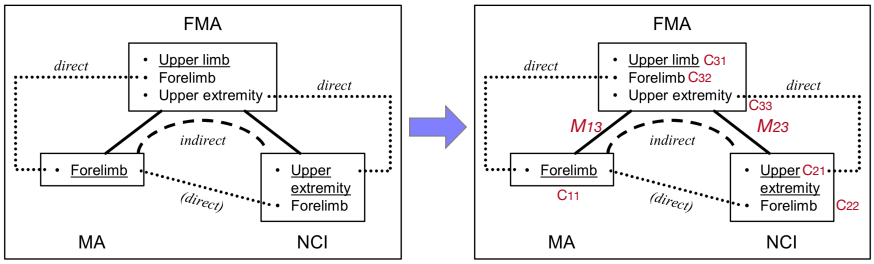


Fig. 2: Left: The original framework for employing external ontology to connect concepts. Right: The embedding framework for employing external ontology to connect concepts

Fig. 2 shows a change of the framework from string equality to the softened idea, where every concept C is represented as a d-dimensional continuous vector $\mathbf{C} \in \mathbb{R}^d$. We assume that there exist some concept pairs (C_1, C_2) involving their synonyms from ontologies O_1 and O_2 will share the same concept C_3 or its synonyms in external ontology O_3. The tuple is labeled as (C_1, C_2, C_3). Then, we introduce two matrices and train them based on these tuples in order to obtain more potential mappings. The loss function is defined as follows.

$$l_{RE} = -\sum_{(C_1,C_2,C_3)\in\mathcal{T}} log f_{RE}(C_1, C_2, C_3) - \sum_{(C_1',C_2',C_3)\in\mathcal{T}'} log(1 - f_{RE}(C_1', C_2', C_3)), \tag{5}$$

where $\mathcal{T}$ is a set of tuples $\{(C_1, C_2, C_3)\}$ generated by the shared assumption, $\mathcal{T}'$ is a set of negative tuples that randomly replace C_1 or C_2. $f_{RE}(C_1, C_2, C_3)$ is a score function defined in Eq. 6 to calculate the score of projected concepts, where $\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3} \in \mathbb{R}^d$ are d-dimensional continuous vectors of concepts C_1, C_2, C_3 from different ontologies, M_{13} and M_{23} are two matrices that project $\mathbf{C_1}$, $\mathbf{C_2}$ into the embedding space of O_3, respectively. We hope that the similar concepts will be projected near their shared concept. Conversely, there existed a semantic distance between dissimilar ones.

$$f_{RE}(C_1, C_2, C_3) = 2 \cdot \frac{1}{1 + e^{(||\mathbf{C_1} * M_{13} - \mathbf{C_3}||_2 + ||\mathbf{C_2} * M_{23} - \mathbf{C_3}||_2)}}. \tag{6}$$

In order to train two matrices better, we maintain all the vectors of concepts in O_3 unchanged and only update the parameters of matrices and concepts from O_1 and O_2. Furthermore, we take advantage of structural relations in O_3 to pre-train the vector representations of concepts, which can adjust semantic distances of concept vectors. As existing KG embedding models face the sparsity problem, we design a loss function

based on cross-entropy to achieve this goal that is defined as follows.

$$l_{PT} = -\sum_{(C_{31},r,C_{32})\in\mathcal{R}} log f_r(C_{31},C_{32}) - \sum_{(C'_{31},r,C'_{32})\in\mathcal{R}'} log(1-f_r(C'_{31},C'_{32})), \quad (7)$$

$$f_r(C_{31},C_{32}) = 2\cdot\frac{1}{1+e^{(||\mathbf{C_{31}}-\mathbf{C_{32}}||_2-\alpha)}}, \quad (8)$$

where $\mathcal{R}$ is a set of relation assertions, involving $\{(C_{31}, \textit{subClassOf}, C_{32})\} \cup (C_{31}, \textit{PartOf}, C_{32})\}$, $\mathcal{R}'$ is a set of negative ones that randomly replace C_{31} or C_{32}. $f_r(C_{31}, C_{32})$ is a score function that measures the score of (C_{31}, r, C_{32}), $\mathbf{C_{31}}$ and $\mathbf{C_{32}}$ are vector representations of concepts C_{31} and C_{32}. Notice that, *subClassOf* and *PartOf* are not equivalent relations, so we utilize a hyper-parameter α to controls the semantic distances of concept vectors.

3.2 View-Embedding Combination

After obtained mappings from different modules, we need to combine them together. A straightforward strategy is collecting all the mappings from these modules and filtering out them with one threshold or stable marriage algorithm. Although this strategy can obtain a high recall in the final alignment, it may also introduce lots of wrong mappings and miss n:m cases about mappings. Therefore, we propose a combination strategy based on mutual assessment.

For convenience, we use OM-L, OM-S, OM-R to represent the alignments generated by lexical-based module, structural-based module, resource-based module, respectively. The concrete procedures are achieved as follows.

Step 1 Merge the mappings from OM-S and OM-R. Their merged result is labeled as OM-SR, in which the similarity of each mapping is selected the large one between OM-S and OM-R.

Step 2 Select the "reliable" mappings of OM-L and OM-SR based on the corresponding thresholds δ_1 and δ_2.

Step 3 Assess these "reliable" mappings from OM-L and OM-SR mutually. For example, if one "reliable" mapping belongs to OM-L and its similarity in OM-SR is lower than threshold δ_3, then we need to remove it. Relatively, if one "reliable" mapping belongs to OM-SR and its similarity in OM-L is lower than threshold δ_4, then this mapping will be removed.

Step 4 Merge assessed mappings from OM-L and OM-SR and generate a final alignment. For each mapping appearing in OM-L and OM-SR at the same time, its similarity is selected the large one.

4 Experiments

To verify the effectiveness of MultiOM, we used Python to implement our approach with the aid of TensorFlow[10] and parse ontologies by OWLAPI[11]. The experiments were conducted on a personal workstation with an Intel Xeon E5-2630 V4 CPU which has 64GB memory and TiTAN XP GPU. Our approach[12] can be downloaded together with the datasets and results.

[10] https://www.tensorflow.org/

[11] http://owlapi.sourceforge.net/

[12] https://github.com/chunyedxx/MultiOM

4.1 Datasets

We collect the biomedical ontologies from Anatomy Track and Large BioMed Track in OAEI[13] (Ontology Alignment Evaluation Initiative), which is an annual campaign for evaluating ontology matching systems that attracts many participants all over the world. Furthermore, this campaign provides uniform test cases and standard alignments for measuring precision, recall and F1-measure for all participating systems.

4.2 Experiment Settings

We select several strategies to construct the baseline systems to verify the effectiveness of our model. The following is the detail construction of strategies in our experiments.

- StringEquiv: It is a string matcher based on string equality applied on local names of entities.
- StringEquiv + Normalization (StringEquiv-N): It employs normalization techniques before execute StringEquiv matcher.
- StringEquiv + Synonym (StringEquiv-S): It extends the synonym of concepts when executing the StringEquiv matcher.
- StringEquiv + Synonym + Reference Ontology (StringEquiv-SR): It introduces external ontologies as bridges to connect concepts based on StringEquiv-S.
- StringEquiv + Synonym + Normalization (StringEquiv-NS): It extends the synonym of concepts when executing the StringEquiv-N.
- StringEquiv + Normalization+ Synonym + Reference Ontology (StringEquiv-NSR): employs normalization techniques before execute StringEquiv-SR.

For MultiOM, we use stochastic gradient descent (SGD) as an optimizer and the configuration of hyper-parameters is listed below: Dimensions of concepts and matrices are set to d=$\{\mathbf{50}, 100\}$ and $d_{\mathcal{M}}$=$\{\mathbf{50}, 100\}$. The mini-batch size of SGD is set to Nbatch=$\{5, \mathbf{10}, 20, 50\}$. We select the learning rate λ among $\{\mathbf{0.01}, 0.02, 0.001\}$ and $\{1, 3, \mathbf{5}, 10\}$ negative triples sampled for each positive triple. The whole training spent 1000 epochs. In lexical-based module, the vector presentations of tokens mainly come from the linkage[14] of the work [11], whose dimension is set to **200**. For some tokens without vector presentations, we initialize them randomly and enforce constrains as $||\mathbf{t_{1i}}||_2 \leq 1$ and $||\mathbf{t_{2j}}||_2 \leq 1$. In resource-view embedding module, we employ TransE [13], ConvE [23] and pre-training function 7 to initialize the vector representations of concepts in external ontologies. α is set to $\{\mathbf{0.01}, 0.05, 0.10\}$ in function 7 for controlling the semantic distances of concept vectors. For negative sampling strategy, we collect all the related structural assertions of concepts. When one concept is selected as a replacer, we retrieve the structural assertions of this concept and execute the replacement based on its relations with the original concept. During this process of replacement, *disjointWith* relations own the highest priority and *subclassOf* relations should be excluded. Finally, the result of MultiOM is generated by the combination strategy, and we set the related thresholds $\delta_1 = 0.8$, $\delta_2 = 0.95$, $\delta_3 = 0.65$, $\delta_4 = 0.3$.

In order to show the effect of our proposed negative sampling, a symbol "-" added to the symbol represented module (or merged one) indicates that this module is not equipped with negative sampling tailored for structural relations.

[13] http://oaei.ontologymatching.org/

[14] https://doi.org/10.5281/zenodo.1173936.

4.3 Evaluation Results

Table 1 lists the matching results of MultiOM compared with baseline systems. We observe that merging more strategies can improve the number of mappings. Although it slightly decreases the precision of alignments, it can increase the recall and F1-measure as a whole. Relatively, MultiOM further improves the recall and F1-measure of alignments because continue vector representations of concepts can obtain more similar concepts and discover more potential mappings. Moreover, the performance of MultiOM is better than MultiOM$^-$ in term of F1-measure. The main reason is that employing structural relations are helpful to distinguish the vector representations of concepts.

Table 1: The comparison of MultiOM with baseline systems

Methods	MA-NCI					FMA-NCI-small				
	Number	Correct	P	R	F1	Number	Correct	P	R	F1
StringEquiv	935	932	0.997	0.615	0.761	1501	1389	0.995	0.517	0.681
StringEquiv-N	992	989	0.997	0.625	0.789	1716	1598	0.995	0.595	0.745
StringEquiv-S	1100	1057	0.961	0.697	0.808	2343	2082	0.974	0.775	0.863
StringEquiv-SR	1162	1094	0.941	0.722	0.817	2343	2082	0.974	0.775	0.863
StringEquiv-NS	1153	1109	0.962	0.732	0.831	2464	2200	0.975	0.819	0.890
StringEquiv-NSR	1211	1143	0.943	0.753	0.838	2467	2203	0.975	0.820	0.891
MultiOM$^-$	1484	1296	0.873	0.855	0.864	2500	2173	0.947	0.809	0.872
MultiOM	1445	1287	0.891	0.849	0.869	2538	2195	0.942	0.817	0.875

Table 2: The results about combining with different embedding modules in Anatomy Track

Methods	Number	Correct	P	R	F1
TFIDF (threshold= 0.8)	985	976	0.991	0.644	0.780
OM-L (threshold= 0.8)	1286	1175	0.914	0.775	0.839
OM-S^- (threshold= 0.95)	1836	1109	0.604	0.732	0.662
OM-S (threshold= 0.95)	1189	1097	0.923	0.724	0.811
OM-R (Random initialization, threshold= 0.95)	709	680	0.959	0.449	0.661
OM-R (TransE, threshold= 0.95)	22	4	0.182	0.003	0.005
OM-R (ConvE, threshold= 0.95)	835	790	0.946	0.521	0.672
OM-R (loss function 7, threshold= 0.95)	833	789	0.948	0.520	0.672
OM-RS^- (threshold= 0.95)	1271	1147	0.902	0.757	0.823
OM-RS (threshold= 0.95)	1237	1138	0.920	0.751	0.827
MultiOM$^-$	1484	1296	0.873	0.855	0.864
MultiOM	1445	1287	0.891	0.849	0.869

Table 2 shows the results of different combination with embedding-view modules. Overall, merge more embedding modules, the performances of alignments are better. For lexical-view module, softened TF-IDF (denoted as OM-L) is better than original TF-IDF in terms of F1-measure because continuous vectors representing tokens can provide more semantic information than single strings for calculating the similarity of concepts. For resource-view embedding module (denoted as OM-R), ConvE and our pre-training function are better than random initialization because both of them can utilize structural relations to adopt vector representations of concepts in the semantic

space. Nevertheless, compared with 20 minutes spent in function 7, ConvE took nearly 24 hours to obtain the vector presentations of concepts. Notice that, it is not suitable for TransE to pre-train the vector presentations of concepts. We analyze that sparse structural relations of ontologies and its simplified score function limit its capability. Overall, we observe that employing new negative sampling strategy in embedding-view modules (i.e., OM-S, OM-RS, MultiOM) is helpful to improve the quality of alignments further in terms of precision and F1-measure.

Table 3 lists the comparison of MultiOM with OAEI 2018 top-ranked systems based on feature engineering and representation learning. Preliminary result shows that MultiOM can be competitive with several promising matching systems (e,g, FCAMapX, XMap) in terms of F1-measure. Nevertheless, there still exists a gap compared with the best systems (e.g., AML, SCBOW + DAE (O)). We analyze that lexical-based module and simplified combination strategy may become the main bottlenecks of MultiOM. Benefited from thesauruses (e.g., UMLS) and optimized combination strategy, most top-ranked systems can obtain better performances of OM tasks. In addition, most systems (e.g., AML, LogMap) employ alignment debugging techniques, which is helpful to improve the quality of alignment further. But we do not employ these techniques in the current version. We leave these issues in our future work.

Table 3: The comparison of MultiOM with OAEI 2018 top-ranked systems

Methods	MA-NCI					FMA-NCI-small				
	Number	Correct	P	R	F1	Number	Correct	P	R	F1
AML	1493	1418	0.95	0.936	0.943	2723	2608	0.958	0.910	0.933
SCBOW + DAE(O)	1399	1356	0.969	0.906	0.938	2282	2227	0.976	0.889	0.930
LogMapBio	1550	1376	0.888	0.908	0.898	2776	2632	0.948	0.902	0.921
POMAP++	1446	1329	0.919	0.877	0.897	2414	2363	0.979	0.814	0.889
XMap	1413	1312	0.929	0.865	0.896	2315	2262	0.977	0.783	0.869
LogMap	1387	1273	0.918	0.846	0.880	2747	2593	0.944	0.897	0.920
SANOM	1450	1287	0.888	0.844	0.865	–	–	–	–	–
FCAMapX	1274	1199	0.941	0.791	0.859	2828	2681	0.948	0.911	0.929
MultiOM	**1445**	**1287**	**0.891**	**0.849**	**0.869**	**2538**	**2195**	**0.942**	**0.817**	**0.875**

5 Conclusion and future work

In this paper, we presented an alternative OM framework called MultiOM, in which different loss functions were designed based on cross-entropy to model different views among ontologies and learn the vector representations of concepts. We further proposed a novel negative sampling skill tailored for structural relations, which could obtain better vector representations of concepts. We implemented our method and evaluated it on real-world biomedical ontologies. The preliminary result indicated that MultiOM was competitive with several OAEI top-ranked systems in terms of F1-measure.

In the future work, we will explore following research directions: (1) As candidate mappings and tuples are not enough, we will extend MultiOM to an iterative framework. (2) Recently, Zhang et al. [17] presented combination strategies for entity alignment based on embedding techniques. Incorporating these combination strategies into MultiOM may facilitate improving the quality of mappings. (3) Senior symbolic reasoning techniques (e.g., incoherent checking) could be served for training process and alignment generation. We will merge them into MultiOM for improving its performances.

References

1. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Science, 2013.
2. Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. Ontology matching: A literature review. *Expert Syst. Appl.*, 42(2):949–971, 2015.
3. Ian Harrow, Ernesto Jiménez-Ruiz, Andrea Splendiani, Martin Romacker, Peter Woollard, Scott Markel, Yasmin Alam-Faruque, Martin Koch, James Malone, and Arild Waaler. Matching disease and phenotype ontologies in the ontology alignment evaluation initiative. *Journal of Biomedical Semantics*, 8(1):1–13, 2017.
4. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-Based and Scalable Ontology Matching. In *ISWC*, pages 273–288, 2011.
5. Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F Cruz, and Francisco M Couto. The AgreementMakerLight Ontology Matching System. In *OTM Conferences*, pages 527–541, 2013.
6. Mengyi Zhao, Songmao Zhang, Weizhuo Li, and Guowei Chen. Matching biomedical ontologies based on formal concept analysis. *Journal of Biomedical Semantics*, 9(1):11, 2018.
7. Warith Eddine Djeddi and Mohamed Tarek Khadir. A Novel Approach Using Context-Based Measure for Matching Large Scale Ontologies. In *DaWaK*, pages 320–331, 2014.
8. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436, 2015.
9. Chuncheng Xiang, Tingsong Jiang, Baobao Chang, and Zhifang Sui. ERSOM: A Structural Ontology Matching Approach Using Automatically Learned Entity Representation. In *EMNLP*, pages 2419–2429, 2015.
10. Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. DeepAlignment: Unsupervised Ontology Matching with Refined Word Vectors. In *NAACL*, pages 787–798, 2018.
11. Prodromos Kolyvakis, Alexandros Kalousis, Barry Smith, and Dimitris Kiritsis. *Journal of Biomedical Semantics*, 9(1):21, 2018.
12. Lucy Wang, Chandra Bhagavatula, Mark Neumann, Kyle Lo, Chris Wilhelm, and Waleed Ammar. Ontology alignment in the biomedical domain using entity definitions and context. In *BioNLP*, pages 47–55, 2018.
13. Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *NeurIPS*, pages 2787–2795, 2013.
14. Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. Iteratively learning embeddings and rules for knowledge graph reasoning. In *WWW*, pages 2366–2377, 2019.
15. George A Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
16. Meng Qu, Jian Tang, Jingbo Shang, Xiang Ren, Ming Zhang, and Jiawei Han. An attention-based collaboration framework for multi-view network representation learning. In *CIKM*, pages 1767–1776, 2017.
17. Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Multi-view Knowledge Graph Embedding for Entity Alignment. In *Proceedings of IJCAI*, 2019.
18. Miguel Ángel Rodríguez-García, Georgios V Gkoutos, Paul N Schofield, and Robert Hoehndorf. Integrating phenotype ontologies with PhenomeNET. *Journal of Biomedical Semantics*, 8(1):58, 2017.
19. Michelle Cheatham and Pascal Hitzler. String Similarity Metrics for Ontology Alignment. In *ISWC*, pages 294–309, 2013.
20. Yuanzhe Zhang, Xuepeng Wang, Siwei Lai, Shizhu He, Kang Liu, Jun Zhao, and Xueqiang Lv. Ontology Matching with Word Embeddings. In *CCL*, pages 34–45. 2014.
21. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS*, pages 3111–3119, 2013.
22. Songmao Zhang and Olivier Bodenreider. Experience in Aligning Anatomical Ontologies. *International Journal on Semantic Web and Information Systems*, 3(2):1–26, 2007.
23. Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*, pages 1811–1818, 2018.

Identifying Mappings among Knowledge Graphs by Formal Concept Analysis

Guowei Chen[1,2] and Songmao Zhang[2]

[1] University of Chinese Academy of Sciences, Beijing, P.R. China
[2] Institute of Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, P.R. China
chenguowei17@mails.ucas.ac.cn, smzhang@math.ac.cn

Abstract. Formal Concept Analysis (FCA) is a well-developed mathematical model for clustering individuals and structuring concepts. In one of our previous studies, we proposed to incrementally match classes and properties across complex biomedical ontologies based on FCA. We intend to apply the approach to matching knowledge graphs (KGs) and this paper reports a preliminary result. Compared with ontologies which model the schema knowledge of classes, KGs are much larger and focus on instances and their properties. We build three token-based formal contexts for classes, properties, and instances to describe how their names/labels share lexical tokens, and from the concept lattices computed, lexical mappings can be extracted across KGs. An evaluation on the 9 matching tasks of OAEI Knowledge Graph Track shows that our system obtains the highest recall in class, property, instance, and overall matching over the seven systems participated in the track in OAEI 2018. Additionally, our system is able to identify cases when one entity in a KG does not have any correspondence in another KG. Based on the lexical instance mappings, we further construct a property-based formal context to identify commonalities among properties in a structural way, which indicates a promising direction for taking full advantage of the knowledge within KGs.

Keywords: knowledge graph · formal concept analysis · ontology matching

1 Introduction

Ontologies serve as the foundation of the Semantic Web by defining basic classes and their structures that constitute various domain knowledge, thus can be used to semantically annotate the Web resources. Ontology matching (OM) techniques [1] have been developed to detect the correspondence among diverse yet overlapping ontologies so that search engines and applications can understand the equivalence on the Web as well as mismatches. Since Google invented the

notion of Knowledge Graph (KG) and made its own system in 2002, and with the prevailing of the TransE series algorithms [2,3] for embedding KGs in a numerical way, the Semantic Web has evolved into the KG time. Soon the OM community realized the inevitable of identifying semantic connections among KGs. Started in 2018, the annual OAEI competition [3] presents a KG track where 9 KGs in the category of Games, Comins, and TV&Books, respectively, yield a total of 9 pairwise matching tasks [5,6]. Seven OM systems were able to participate in the KG track in 2018, including the well-known AML [7], LogMap family [8], POMAP++ [9], Holontology [10], and DOME [11].

By design, both ontologies and KGs have classes, properties and instances. Ontologies primarily model the schema knowledge of classes whereas KGs are much larger and mostly describe instances and their properties. This means that techniques for mapping KGs focus more on instance matching [12]. In one of our previous studies [18,19,20], we proposed the FCA-Map system that incrementally matches classes and properties across complex biomedical ontologies based on Formal Concept Analysis (FCA). FCA is a well-developed mathematical model for clustering individuals and structuring concepts [14]. The purpose of FCA-Map is to push the envelop of the FCA formalism in exploring as much knowledge as possible within ontologies, including class names, subclass relations, part-whole relations, disjointedness, and other logical axioms. In this paper, we intend to apply the approach to matching knowledge graphs and a preliminary result is reported.

Concretely, based on the rationale of lexical matching in FCA-Map, we construct three token-based formal contexts for classes, properties, and instances, respectively, to describe how their names/labels share lexical tokens. The derived formal concept lattices represent the clustering of classes/properties/instances by names, and thus lexical mappings can be extracted across KGs. An evaluation on the OAEI KG Track shows that, when compared with the seven OAEI 2018 participants, our system obtains the highest recall and comes second in F-measure in terms of average performances on 9 tasks. In addition, our system can identify most of the null mappings provided in the OAEI gold standard for entities that do not have any correspondence in another KG. Based on the lexical mappings, we further build a structural formal context to describe how properties across KGs have common in linking the same instances. The mappings identified solely by structural matching indicate a promising direction for taking full advantage of the knowledge within KGs.

Although FCA has been applied to modeling KGs [13], to the best of our knowledge, this is a first attempt to identify the correspondence among KGs by a FCA-based approach. In Section 2 of the paper, we will present the lexical matching part and its evaluation on the OAEI KG Track. A first step of structural matching is described in Section 3, and our on-going work is discussed in Section 4 at last.

[3] http://oaei.ontologymatching.org/

2 Identifying lexical mappings between KGs

FCA is a principled approach of deriving a concept hierarchy from a collection of objects and their attributes. The fundamental notions are *formal context* and *formal concept*, and the former is defined as a binary table $\mathbb{K} := (G, M, I)$, where G is a set of objects as rows, M a set of attributes as columns, and I a binary relation between G and M in which $(g, m) \in I$ reads object g has attribute m , generally represented by "$\times$" in the table cell. A *formal concept* of context $\mathbb{K}$ is a pair (A, B) consisting of a subset of objects $A \subseteq G$ and a subset of attributes $B \subseteq M$ such that B equals all the attributes common to objects in A and at the same time, A equals the set of objects that have all the attributes in B. The subconcept-superconcept relation can be defined as: $(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_1 \supseteq B_2)$, leading to a lattice structure of formal concepts.

For the instances in two KGs, we use the following example to illustrate the construction of token-based formal context, the derivation of concept lattice and the extraction of instance mappings. The similar process applies to the classes and properties in two KGs.

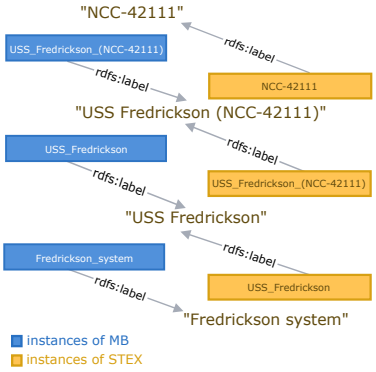
Example 1. Given two KGs *memory-beta* (MB), *stexpanded* (STEX) from OAEI 2018, the left of Fig. 1 shows some instances and their label strings. Note that one string can be shared by instances across KGs, as listed on the right of Fig. 1. We extract names and labels of all instances in the two KGs and separate the tokens in them through normalization techniques [17]. As shown in Fig. 2 on the left, the token-based formal context is constructed with each string as an object, each token as an attribute, and the cell in the context marked when the string contains the token. The gray area in the table presents a formal concept indicating the duality between its objects and attributes, i.e., the subset of tokens are identified to co-exist solely in the two strings.

From the token-based formal context, formal concepts and their lattice structure can be derived automatically, as shown on the right of Fig. 2, where each node represents a formal concept and the line denotes the subconcept-superconcept relation from the lower to the upper node [4]. For identifying mappings, we pay attention to formal concepts that contain exactly two strings relevant to instances across KGs. Take for example the gray node on the right of Fig. 2 which corresponds to the gray area in the context on the left. Four instance mappings can be extracted from this formal concept:

⟨MB:`USS_Fredrickson`, STEX:`USS_Fredrickson`⟩
⟨MB:`USS_Fredrickson_(NCC-42111)`, STEX:`USS_Fredrickson_(NCC-42111)`⟩
⟨MB:`USS_Fredrickson`, STEX:`USS_Fredrickson_(NCC-42111)`⟩
⟨MB:`USS_Fredrickson_(NCC-42111)`, STEX:`USS_Fredrickson`⟩

The first two are exact matches and the latter partial matches.

[4] For the sake of efficiency, we use the Galois Sub-Hierarchy (GSH) [15] which preserves solely the necessary elements of the lattice and implement the *Hermes*[16] algorithm for computing the lattice.



String	Instances
uss fredrickson	MB:USS_Fredrickson STEX:USS_Fredrickson
uss fredrickson (ncc-42111)	MB:USS_Fredrickson_(NCC-42111) STEX:USS_Fredrickson_(NCC-42111)
ncc-42111	MB:NCC-42111
fredrickson system	MB:Fredrickson_system

Fig. 1. Left: An RDF graph representation of part of two KGs in *Example 1*. **Right:** Strings and the instances (can be across KGs) having them as labels.

	uss	fredrickson	ncc	42111	system
uss fredrickson	×	×			
uss fredrickson (ncc-42111)	×	×	×	×	
ncc-42111			×	×	
fredrickson system		×			×

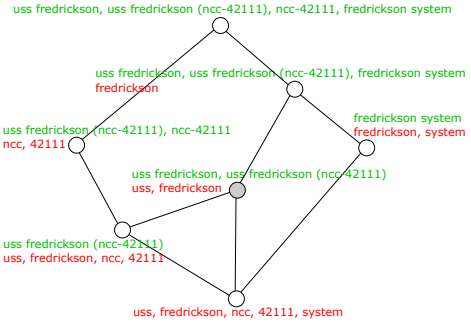


Fig. 2. Left: The token-based formal context for instances in *Example 1*. **Right:** The derived formal concept lattice.

There are 9 knowledge graphs in the OAEI KG Track, as listed in Table 1, and on its corresponding 9 KG matching tasks, we evaluate our FCA-based lexical matching approach. The results are shown in Fig. 3 according to the gold standard[5] and evaluation tool[6] provided by OAEI 2018. One can see that our approach is able to achieve high performances in recall, and the quality of class mappings is better than that of property mappings which is then better than instance mappings while at the same time the number of mappings identified for class, property and instance increases.

A comparison with the seven OAEI 2018 KG Track participants is listed in Table 2. Again, our approach favors recall and ranks the first in average over 9 tasks for class, property, instance and overall matching. Moreover, our approach obtains the second best F-measures in all matching types, indicating that a bal-

[5] https://github.com/sven-h/dbkwik/tree/master/e_gold_mapping_interwiki/gold

[6] http://oaei.ontologymatching.org/2018/results/knowledgegraph/kg_track_eval.zip

Table 1. An overview of 9 knowledge graphs of the OAEI KG Track

KG	Category	#Class	#Property	#Instance
RuneScape Wiki (runescape)	Games	106	1,998	200,605
Old School RuneScape Wiki (oldschoolrunescape)	Games	53	488	38,563
DarkScape Wiki (darkscape)	Games	65	686	19,623
Marvel Database (marvel)	Comics	2	99	56,464
Hey Kids Comics Wiki (heykidscomins)	Comics	181	1,925	158,234
DC Database (dc)	Comics	5	177	128,495
Memory Alpha (memory-alpha)	TV	0	326	63,240
Star Trek Expanded Universe (expanded)	TV	3	201	17,659
Memory Beta (memory-beta)	Books	11	413	63,223

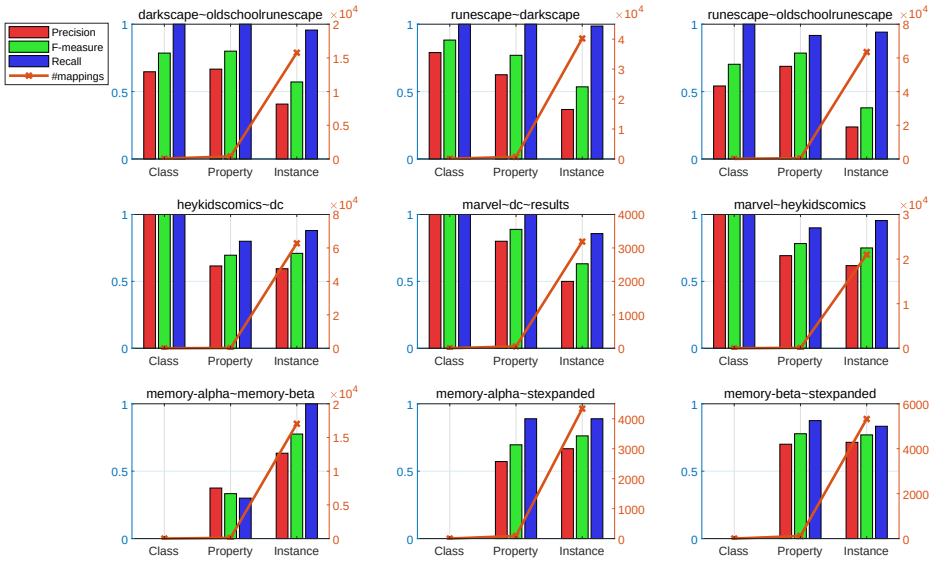


Fig. 3. The results of FCA-based KG matching. Charts in the same row are about the same category, i.e., Games, Comics, and TV&Books. In each chart, the bars show precision, F-measure and recall of each task, whereas the lines show the number of mappings identified by our approach.

ance can be achieved between quality and quantity. Overall, the DOME system [11] stands out by having the best precision and F-measure in both property matching and instance matching for most cases, followed by Holontology [10] which ranks the first in overall precision.

Table 2. Comparing with OAEI 2018 KG Track participants by average performance over 9 matching tasks, where # stands for the number of tasks that the system is able to generate non-empty alignments, and *Size* the average number of generated mappings.

System	#	Class				Property				Instance				overall			
		Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
AML	5	11.6	**0.85**	0.64	0.51	0.0	0.00	0.00	0.00	82380.9	0.16	0.23	0.38	**102471.1**	0.19	0.23	0.31
POMAP++	9	15.1	0.79	0.74	0.69	0.0	0.00	0.00	0.00	0.0	0.00	0.00	0.00	16.9	0.79	0.14	0.08
Holontology	9	16.8	0.80	**0.83**	0.87	0.0	0.00	0.00	0.00	0.0	0.00	0.00	0.00	18.8	**0.80**	0.17	0.10
DOME	9	16.0	0.73	0.73	0.73	207.3	**0.86**	**0.84**	0.81	15688.7	**0.61**	**0.61**	0.61	15912.0	0.68	**0.68**	0.67
LogMap	7	21.7	0.66	0.77	0.91	0.0	0.00	0.00	0.00	**97081.4**	0.08	0.14	0.81	97104.8	0.09	0.16	0.64
LogMapBio	9	22.1	0.68	0.81	**1.00**	0.0	0.00	0.00	0.00	0.0	0.00	0.00	0.00	24.1	0.68	0.19	0.11
LogMapLt	6	22.0	0.61	0.72	0.87	0.0	0.00	0.00	0.00	82388.3	0.39	0.52	0.76	88893.1	0.42	0.49	0.60
Our System	9	**22.7**	0.68	0.81	**1.00**	**250.9**	0.64	0.74	**0.86**	25903.9	0.39	0.55	**0.95**	26177.4	0.45	0.61	**0.93**

Table 3. Null mappings identified by our system, where *Gold* stands for the number of null mappings in the gold standard.

KG matching task	Class			Property			Instance		
	Gold	In gold	Not in gold	Gold	In gold	Not in gold	Gold	In gold	Not in gold
darkscape oldschoolrunescape	7	6	22	6	6	455	38	34	25,032
runescape darkscape	5	5	38	10	10	1,339	13	3	107,941
runescape oldschoolrunescape	4	3	53	8	8	1,611	37	11	115,061
heykidscomics dc	13	12	123	10	8	1,512	53	40	156,744
marvel dc	3	3	0	12	11	143	65	56	164,543
marvel heykidscomics	10	4	128	10	8	1,517	42	38	160,706
memory-alpha memory-beta	11	11	1	10	7	511	49	42	92,334
memory-alpha stexpanded	3	3	1	11	11	339	60	57	69,823
memory-beta stexpanded	14	14	0	12	11	369	55	51	67,848

The gold standard of OAEI KG Track contains not only 1:1 mappings but also cases where one entity in a KG is matched to "null" in the other KG. They represent the uniqueness of classes, properties and instances to one knowledge base with respect to another, which is complementary to 1:1 and complex mappings in revealing the whole picture of the relationship between two systems. We call them *null* mappings, and the OAEI evaluation takes them into account solely for calculating false positives in 1:1 mappings. By taking advan-

tage of the inherent feature of the FCA formalism, our system is able to identify such null mappings. When a formal concept in the derived lattice contains strings solely from one entity in a KG, the corresponding entity contributes to a null mapping. As shown in Table 3, there are 571 null mappings in the gold standard and our system has successfully detected 473 of them, accounting for 83%, as exemplified by ⟨darkscape:`Room`, oldschoolrunescape:`null`⟩ for class null mapping, ⟨marvel:`null`, dc:`runtime`⟩ for property, and ⟨memory-beta:`Victoria`, stexpanded:`null`⟩ for instance. At the same time, a large number of null mappings identified are not in the gold standard, and their validity needs further investigation as the gold standard is only partial as reported by OAEI.

3 Identifying structural mappings between KGs

We call the obtained lexical mappings anchors, based on which we can build formal contexts from the structural knowledge in KGs so as to extract additional mappings. A KG can be seen as an RDF graph where the vertex generally represents a class or an instance and the edge a property from one instance to another, or a type relation from an instance to a class. For given two KGs, a property-based formal context is constructed by taking properties from two KGs as objects, and pairing the lexical instance anchors across KGs as attributes. When a property is used to link two instances in an anchor pair, the corresponding cell in the formal context is marked. After the lattice is derived, if a formal concept contains solely two properties from two KGs, respectively, they can be extracted as a structural mapping. Again, in the following we use an example to illustrate the matching process.

Example 2. Given two KGs *memory-alpha* (MA), *memory-beta* (MB) from OAEI 2018, a part of their *(subject, predicate, object)* (SPO triples) are listed in Table 4.

Table 4. Some SPO triples from two KGs MA and MB.

subject	**predicate**	**object**
MA:`Rules_of_Acquisition_(episode)`	MA:`wsstoryby`	MA:`Hilary_J._Bader`
MA:`Rules_of_Acquisition_(episode)`	MA:`wsteleplayby`	MA:`Ira_Steven_Behr`
MA:`Battle_Lines_(episode)`	MA:`wsstoryby`	MA:`Hilary_J._Bader`
MA:`Battle_Lines_(episode)`	MA:`wsteleplayby`	MA:`Richard_Danus`
MA:`Paradise_Lost_(episode)`	MA:`wsteleplayby`	MA:`Robert_Hewitt_Wolfe`
MB:`Rules_of_Acquisition_(episode)`	MB:`story`	MB:`Hilary_J._Bader`
MB:`Rules_of_Acquisition_(episode)`	MB:`teleplay`	MB:`Ira_Steven_Behr`
MB:`The_Nagus`	MB:`teleplay`	MB:`Ira_Steven_Behr`
MB:`Battle_Lines_(episode)`	MB:`story`	MB:`Hilary_J._Bader`
MB:`Paradise_Lost_(episode)`	MB:`teleplay`	MB:`Robert_Hewitt_Wolfe`

Some lexical instance anchors between MA and MB are as follow:

$a = \langle$MA:`Battle_Lines_(episode)`, MB:`Battle_Lines_(episode)`$\rangle$
$b = \langle$MA:`Hilary_J._Bader`, MB:`Hilary_J._Bader`$\rangle$
$c = \langle$MA:`Ira_Steven_Behr`, MB:`Ira_Steven_Behr`$\rangle$
$d = \langle$MA:`Paradise_Lost_(episode)`, MB:`Paradise_Lost_(episode)`$\rangle$
$e = \langle$MA:`Rules_of_Acquisition_(episode)`, MB:`Rules_of_Acquisition_(episode)`$\rangle$
$f = \langle$MA:`Richard_Danus`, MB:`Richard_Danus`$\rangle$
$g = \langle$MA:`Robert_Hewitt_Wolfe`, MB:`Robert_Hewitt_Wolfe`$\rangle$
$h = \langle$MA:`The_Nagus`, MB:`The_Nagus`$\rangle$.

	(d,g)	(e,c)	(e,b)	(a,b)	(a,f)	(h,c)
MA:`wsteleplayby`	×	×			×	
MB:`teleplay`	×	×				×
MA:`wsstoryby`			×	×		
MB:`story`			×	×		

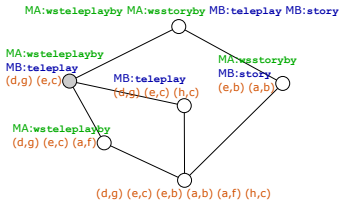


Fig. 4. Left: The structural formal context for properties in *Example 2*. **Right:** The derived formal concept lattice.

Table 5. The property mappings solely identified structurally between two KGs MA and MB.

	Property mapping
Those in the gold standard	$\langle$MA:`relative`, MB:`otherRelatives`$\rangle$ $\langle$MA:`wsteleplayby`, MB:`teleplay`$\rangle$
Those not in the gold standard	$\langle$MA:`wsstoryby`, MB:`story`$\rangle$ $\langle$MA:`prev`, MB:`before`$\rangle$ $\langle$MA:`next`, MB:`after`$\rangle$ $\langle$MA:`relative`, MB:`grandparents`$\rangle$ $\langle$MA:`abreadby`, MB:`narrator`$\rangle$

The constructed property-based formal context is presented on the left in Fig. 4 and the lattice derived on the right. As shown by the gray area, a property mapping $\langle$MA:`wsteleplayby`, MB:`teleplay`$\rangle$ is identified by structural knowledge rather than by names. For the matching task between KGs MA and MB, 7 property mappings are detected solely by the structural matching, as listed in Table 5, of which 2 are true positives. Note that the OAEI 2018 KG gold standard is declared to be only partial, and the lower part of Table 5 shows promising candidates. With these additional structural mappings, the precision, F-measure and recall for the property task have all increased compared with the lexical matching step, as shown by Fig. 5.

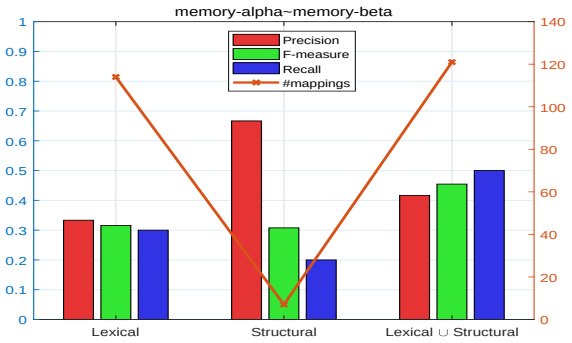


Fig. 5. Evaluation of the additional structural mappings between properties of two KGs MA and MB.

On the other hand, the structural property matching does not affect the performance of the other 8 tasks, either because the mappings found are not in the gold standard or none mappings are found at all. Note that as shown by Fig. 3, these 8 property tasks have already obtained a higher performance compared with the MA-MB task at the lexical matching step. To further improve, comprehensive ways shall be explored to augment the structural formal contexts with extended knowledge in KGs.

4 Discussion and conclusions

This paper reports an on-going study of constructing multiple FCA structures for the purpose of matching knowledge graphs. Its lexical matching part already receives the best recall and the second best F-measure in class, property, instance, and overall matching for the OAEI 2018 KG Track tasks, revealing the advantage of our FCA-based approach. Moreover, our system has identified 83% of null mappings provided in the OAEI gold standard. All these come from the inherent capability of FCA formalism in detecting commonalities among individuals and accordingly forming concepts and classifying them in a lattice structure. For the structural matching, we have realized a property-based lattice from the knowledge of property linking one instance to another in KGs. Obviously, further an instance-based lattice shall be computed similarly to identify structural instance mappings. Moreover, the knowledge of instance belonging to class in KGs can be used as well to explore commonalities among instances. As a matter of fact, we are developing an iterative framework so as to perform class, property, and instance matching in an augmented way until no further matches can be found.

Our previous system FCA-Map is for matching ontologies and thus targets classes. Although there are classes in the OAEI KGs, they are much fewer than instances and properties, and basically none schema knowledge is specified. This

says that the structural matching part in FCA-Map cannot be applied directly, and alternative types of formal contexts are being designed targeting instances and properties. In addition to matching, FCA-Map includes a structural validation step to eliminate wrong mappings based on the disjoint axioms in ontologies. When there is no such knowledge in KGs, we shall develop alternative validation strategies so as to ensure the quality of mappings and prevent the mismatches from propagating in the iterative framework.

What is worth noting is that the systems participated in OAEI 2018 are basically ontology matching systems and not specifically tailored for knowledge graph matching. Therefore it is understandable that the performance can be unsatisfactory for some tasks. Nevertheless, systems like DOME still managed to outperform. DOME uses the doc2vec approach to train vector representations for ontology classes and instances based on large texts, so that the similarity among entities can be computed according to the distance of vectors. Such numerical ways of embedding KG entities into a high-dimensional, continuous space are called representation learning, which have already been adopted for matching ontologies, as in [21,22,23]. To compare our FCA-based approach with these works will be of interest, not only by conducting comparative experiments but also exploring the possible combining ways.

Acknowledgements. This work has been supported by the National Key Research and Development Program of China under grant 2016YFB1000902 and the Natural Science Foundation of China under No. 61621003.

References

1. Euzenat, J., & Shvaiko, P. (2013). Ontology matching, 2nd Edition. Heidelberg: Springer.
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems* (pp. 2787-2795).
3. Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12), 2724-2743.
4. Da Silva, J., Revoredo, K., Baiao, F. A., & Euzenat, J. (2018, October). Interactive ontology matching: using expert feedback to select attribute mappings. In *13th ISWC workshop on ontology matching (OM)* (pp. 25-36). No commercial editor..
5. Sven Hertling, Heiko Paulheim: DBkWik: A Consolidated Knowledge Graph from Thousands of Wikis. *International Conference on Big Knowledge 2018*.
6. Alexandra Hofmann, Samresh Perchani, Jan Portisch, Sven Hertling, and Heiko Paulheim. DBkWik: Towards Knowledge Graph Creation from Thousands of Wikis. *International Semantic Web Conference (Posters & Demos) 2017*.
7. Faria, D., Pesquita, C., Balasubramani, B. S., Tervo, T., Carriço, D., Garrilha, R., ... & Cruz, I. F. (2018, December). Results of AML participation in OAEI 2018. In *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop* (p. 125).
8. Jiménez-Ruiz, E., Grau, B. C., & Cross, V. (2018, December). LogMap family participation in the OAEI 2018. In *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop* (p. 187).

9. Laadhar, A., Ghozzi, F., Megdiche Bousarsar, I., Ravat, F., Teste, O., & Gargouri, F. (2018). OAEI 2018 results of POMap++. In *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop* (p. 192).
10. Roussille, P., Megdiche Bousarsar, I., Teste, O., & Trojahn, C. (2018). Holontology: results of the 2018 OAEI evaluation campaign. In *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop* (p. 167).
11. Hertling, S., & Paulheim, H. (2018, December). DOME results for OAEI 2018. In *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop* (p. 144).
12. Pershina, M., Yakout, M., & Chakrabarti, K. (2015, October). Holistic entity matching across knowledge graphs. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 1585-1590). IEEE.
13. Ferré, S., & Cellier, P. (2019). Graph-FCA: An extension of formal concept analysis to knowledge graphs. *Discrete Applied Mathematics*.
14. Ganter, B., & Wille, R. (2012). Formal concept analysis: mathematical foundations. Springer Science & Business Media.
15. Godin, R., & Mili, H. (1993, September). Building and maintaining analysis-level class hierarchies using galois lattices. In *OOPSLA* (Vol. 93, pp. 394-410).
16. Berry, A., Gutierrez, A., Huchard, M., Napoli, A., & Sigayret, A. (2014). Hermes: a simple and efficient algorithm for building the AOC-poset of a binary relation. *Annals of Mathematics and Artificial Intelligence*, 72(1-2), 45-71.
17. Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137.
18. Zhao, M., & Zhang, S. (2016, October). Identifying and validating ontology mappings by formal concept analysis. In *OM@ ISWC* (pp. 61-72).
19. Zhao, M., Zhang, S., Li, W., & Chen, G. (2018). Matching biomedical ontologies based on formal concept analysis. *Journal of biomedical semantics*, 9(1), 11.
20. Chen, G., & Zhang, S. (2018, December). FCAMapX results for OAEI 2018. In *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop* (p. 160).
21. Xiang, C., Jiang, T., Chang, B., & Sui, Z. (2015). Ersom: A structural ontology matching approach using automatically learned entity representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2419-2429).
22. Kolyvakis, P., Kalousis, A., & Kiritsis, D. (2018, June). Deepalignment: Unsupervised ontology matching with refined word vectors. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 787-798).
23. Kolyvakis, P., Kalousis, A., Smith, B., & Kiritsis, D. (2018). Biomedical ontology alignment: an approach based on representation learning. *Journal of biomedical semantics*, 9(1), 21.

Hypernym relation extraction for establishing subsumptions: preliminary results on matching foundational ontologies

Mouna Kamel[1⋆], Daniela Schmidt[2], Cassia Trojahn[1], and Renata Vieira[2]

[1] Institut de Recherche en Informatique de Toulouse, Toulouse, France
{mouna.kamel,cassia.trojahn}@irit.fr
[2] Pontificia Universidade Catolica do Rio Grande do Sul, Porto Alegre
daniela.schmidt@acad.pucrs.br, renata.vieira@pucrs.br

Abstract. This paper presents an approach for matching foundational ontologies involving subsumption relations. The approach relies on extracting hypernym relations from ontology annotations for establishing such kind of correspondences. We report preliminary results on exploiting lexico-syntactic patterns and definitions layout. Experiments were run on DOLCE and SUMO and the generated alignment was evaluated on a manually generated subsumption reference.

1 Introduction

Foundational ontologies describe general concepts (e.g., physical object) and relations (e.g., parthood), which are independent of a particular domain. The clarity in semantics and the rich formalization of these ontologies are fundamental requirements for ontology development [5] improving ontology quality. They may also act as semantic bridges supporting interoperability between ontologies [8, 10]. However, the development of different foundational ontologies re-introduces the interoperability problem, as stated in [6]. This paper addresses the problem of matching foundational ontologies.

Early works addressed this problem on different perspectives e.g., discussing their different points of view [14, 16, 9] or providing concept alignments between them [13, 7]. Few works have addressed the automatic matching of this kind of ontologies, such as in [7] where alignments between BFO, DOLCE and GFO were built both with automatic tools and manually, with substantially fewer alignments found by the tools. In fact, current tools fail on correctly capturing the semantics behind the ontological foundational concepts, what requires deeper contextualization of the concepts. Besides that, the task requires the identification of other relations than equivalences, such as subsumption and meronym. Few systems are able to discover other relations than equivalence (e.g., AML and BLOOM), with few propositions in the literature [19, 20]. We argue here that the knowledge encoded in the ontologies has to be further exploited. In that way, we propose to borrow approaches from relation extraction from text in NLP in order to establish subsumption relations between the ontologies to be matched.

While the approach is not completely new, as NLP techniques are often used to extract knowledge from text, their exploitation in ontology matching brings some novelty.

Relation extraction in ontology matching has been considered in few works. In [15], a supervised method learns patterns of subsumption evidences, while in [1] the approach relies on free-text parts of Wikipedia in order to help detecting different types of relations, even without clear evidence in the input ontologies themselves. Hearst patterns has been adopted in [17] and [18], with the former using them to eliminate noise in matching results. Here, we report preliminary results on exploiting lexico-syntactic patterns from Hearst [4] and evidences of hypernym relation carried out in definitions layout. Experiments were run on DOLCE and SUMO and the generated alignment has been evaluated on a manually generated subsumption reference. The novelty here is to exploit such methods for foundational ontology matching involving subsumption.

2 Proposed approach

Our approach relies on two main steps: (i) hypernym extraction from ontology annotations and (ii) subsumption generation between ontology concepts, as detailed below.

Hypernym extraction The hypernym relation extraction takes as input the ontology annotations as concept definitions (what are common in top-level ontologies). A *definition* attaches a meaning to a term denoting the concept. The term that is to be defined is called the *definiendum*, and the term or action that defines it is called the *definiens*. In the example below, the *definiendum* = "Product" and the *definiens*="An Artifact that is produced by Manufacture and that is intended to be sold". Many linguistic studies show that definitions mostly express one of the main lexical relations e.g., hypernymy, meronymy or synonymy, between *definiens* and *definiendum* [11].

```
<owl:Class rdf:ID= "Product">
  <rdfs:comment> An Artifact that is produced by Manufacture and
  that is intended to be sold.</rdfs:comment>
</owl:Class>
```

Different strategies are exploited for extracting the hypernym relations:

Hypernym relations expressed using definitions layout We focus on cases where the *definiens* starts by expressing an entity (denoted by a term and different from the *definiendum*) which have some properties. In the above example, the entity in the *definiens* is "Artifact" and the property is "that is produced by Manufacture and that is intended to be sold". Thus the *definiendum* (Product) is an *hyponym* of the definiens (Artifact). When no property is expressed, it is usually a synonym relation, as below:

```
<owl:Class rdf:about="#Quale">
  <rdfs:comment> An atomic region. </rdfs:comment>
</owl:Class>
```

Hypernym relations lexically expressed in text annotations OWL class definitions may also be more fine grained exploited, as comment paragraphs may contain well-written text. We then exploit this text using a set of lexico-syntactic patterns from Hearst [4]:

[NP such as {NP ,}* {or|and} NP], [NP like {NP ,}* {or|and} NP], [NP which is an example of NP], [NP including {NP ,}* {or|and} NP], [NP is called NP if], [NP is an NP that].

For instance, the pattern [NP like {NP ,}* {or|and} NP] means that a noun phrase (NP) must be followed by the word "like", which must be followed by an NP or by a list of NPs separated by comma, having before the last NP "or" or "and". When applied on the definition below, the hypernym relations (Self Connected Object, planet), (Self Connected Object, star) and (Self Connected Object, asteroid) can be identified.

```
<owl:Class rdf:about="#AstronomicalBody">
  <rdfs:comment> The Class of all astronomical objects of
     significant size. It includes Self Connected Objects
     like planets, stars, and asteroids ...
 </rdfs:comment>
</owl:Class>
```

Hypernym relations carried out by the concept identifier Hypernym relations may also be identified from modifiers of a head of a compound noun denoting the identifier of the OWL class. In the example above, the hypernym relation (astronomical body, body) can be identified thanks to this strategy.

Subsumption generation Having extracted all the hypernym relations from both ontologies to be matched, we verify if the terms appearing as hyponyms and hypernyms denote concepts in the ontologies. In the example above, as the alignment is directional, "Product" denotes a concept in the source ontology and "Artifact" in the target ontology, hence this hypernym pair is kept.

3 Experiments

Material and methods We used the foundational ontologies DOLCE [3][1], an ontology of *particulars* which aims at capturing the ontological categories underlying human commonsense; and SUMO [12][2], an ontology of particulars and universals. The reference alignment involving 41 subsumption correspondences comes from [13]. The approach has been implemented with GATE: to extract concepts and their associated comments from the ontology OWL file and restructuring them according to an XML format; to identify terms using first the TermoStat term extractor, and then expanding the recognition of terms using JAPE rules (for instance, the sequence made of a TermoStat term preceded or followed by adjectives, constitutes a new term); to annotate the XML corpus with different NLP tools (ANNIE Tokenizer, Stanford POS, Stanford parser, Gazeteer of identified terms); and to identify hypernym relations.

[1] http://www.loa.istc.cnr.it/old/DOLCE.html
[2] https://github.com/ontologyportal/sumo

Results and discussion Table 1 shows the results of each strategy and their combination. As somehow expected, patterns are very precise while head modifier provides good results in terms of recall with respect to the other strategies. Comparing the approach to the OAEI 2018 matchers[3] (Table 2), besides the fact that we do not distinguish subsumption and equivalence relations when computing precision and recall, no matcher were able to find the correspondences. From the 41 reference correspondences, only one correspondence refers to similar terms (`dolce:geographical-object` and `sumo:GeographicArea`) and 5 of them could be found via a head modifier method (e.g., `dolce:organization` and `sumo:PoliticalOrganization`). In order to see how close the generated alignments were to the reference, we have calculated the relaxed precision and recall [2], that measure the closeness of the results to the reference. While the results of our approach are not that close to the reference, in terms of recall we obtain results similar than the relaxed recall for all matchers.

Combination			Layout			Patterns			Head modifier			Layout+patterns		
P	F	R	P	F	R	P	F	R	P	F	R	P	F	R
.27	.23	.20	.18	.13	.10	1,00	.05	.03	.32	.20	.15	.22	.16	.13

Table 1. Results of the different relation extraction strategies.

System	**Classical**			**Relaxed**		
	P	F	R	P	F	R
M1	.00	.00	.00	.00	.00	.00
M2	.00	.00	.00	.33	.18	.15
M3	.00	.00	.00	.39	.27	.21
M4	.00	.00	.00	.77	.34	.21
M5	.00	.00	.00	.32	.25	.17
M6	.00	.00	.00	.28	.14	.12
M7	.00	.00	.00	.57	.31	.21
M8	.00	.00	.00	.50	**.42**	.21
Proposed approach	.27	**.23**	.20	.28	.28	.29

Table 2. Classical and relaxed precision (P), recall (R) and F-measure (F) of the proposed approach and matchers.

4 Conclusions

We have reported here preliminary results on exploiting symbolic hypernym relation extraction approaches for generating subsumption correspondences between foundational ontologies. This task is still a gap in the field and the initial results presented here can be improved in different ways. First of all, we plan to improving the relation extraction by (i) extending the list of lexico-syntactic patterns, (ii) exploiting syntactic analysis of the text and treating anaphores, and (iii) using background resources such as DBpedia, BabelNet (in particular top level layers of these resources). We also plan to combine relation extraction strategies with matching strategies (structural) and word embeddings, as well as to work on other lexical relations like meronymy. Finally, we plan to apply the approach on domain ontologies.

[3] The aim here is not to evaluate the matching systems themselves, for that reason, their names have been anonymized.

Acknowledgments We warmly thank D. Oberle for sending us all the generated alignments between SUMO and DOLCE-Lite.

References

1. E. Beisswanger. Exploiting relation extraction for ontology alignment. In *Proceedings of the International Semantic Web Conference*, pages 289–296, 2010.
2. M. Ehrig and J. Euzenat. Relaxed precision and recall for ontology matching. In *Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies*, 2005.
3. A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening Ontologies with DOLCE. In *Proceedings of the 13th Conference on Knowledge Engineering and Knowledge Management*, pages 166–181, 2002.
4. M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545, 1992.
5. C. Keet. The use of foundational ontologies in ontology development: An empirical assessment. In *Proceedings of the Extended Semantic Web Conference*, pages 321–335, 2011.
6. Z. Khan and C. Keet. Addressing issues in foundational ontology mediation. In *Proceedings of the Conference on Knowledge Engineering and Ontology Development*, pages 5–16, 2013.
7. Z. Khan and C. Keet. The Foundational Ontology Library ROMULUS. In *Proceedings of the 3rd International Conference on Model and Data Engineering*, pages 200–211, 2013.
8. V. Mascardi, A. Locoro, and P. Rosso. Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation. *Knowledge and Data Engineering*, 22(5):609–623, 2010.
9. L. Muñoz and M. Grüninger. Verifying and mapping the mereotopology of upper-level ontologies. In *Proceedings of the International Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 31–42, 2016.
10. J. C. Nardi, R. de Almeida Falbo, and J. P. A. Almeida. Foundational ontologies for semantic integration in EAI: A systematic literature review. In *Proceedings of the 12th IFIP WG Conference on e-Business, e-Services, and e-Society, I3E*, pages 238–249, 2013.
11. R. Navigli, P. Velardi, and J. M. Ruiz-Martínez. An annotated dataset for extracting definitions and hypernyms from the web. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2010.
12. I. Niles and A. Pease. Towards a Standard Upper Ontology. In *Proceedings of the Conference on Formal Ontology in Information Systems*, pages 2–9, 2001.
13. D. Oberle, A. Ankolekar, P. Hitzler, P. Cimiano, M. Sintek, M. Kiesel, B. Mougouie, S. Baumann, S. Vembu, M. Romanelli, and Buitelaar. DOLCE Ergo SUMO: On Foundational and Domain Models in the SmartWeb Integrated Ontology. *Web Semantics*, 5(3):156–174, 2007.
14. A. Seyed. BFO/DOLCE Primitive Relation Comparison. In *Nature Proceedings*, 2009.
15. V. Spiliopoulos, G. A. Vouros, and V. Karkaletsis. On the discovery of subsumption relations for the alignment of ontologies. *Journal of Web Semantics*, 8(1):69 – 88, 2010.
16. L. Temal, A. Rosier, O. Dameron, and A. Burgun. Mapping BFO and DOLCE. In *Proceedings of the World Congress on Medical Informatics*, pages 1065–1069, 2010.
17. W. R. van Hage, S. Katrenko, and G. Schreiber. A method to combine linguistic ontology-mapping techniques. In *International Semantic Web Conference*, pages 732–744, 2005.
18. R. Vazquez and N. Swoboda. Combining the semantic web with the web as background knowledge for ontology mapping. In *Meaningful Internet Systems*, pages 814–831, 2007.
19. A. Vennesland. Matcher composition for identification of subsumption relations in ontology matching. In *Proceedings of the Conference on Web Intelligence*, pages 154–161, 2017.
20. N. Zong, S. Nam, J.-H. Eom, J. Ahn, H. Joe, and H.-G. Kim. Aligning ontologies with subsumption and equivalence relations in linked data. *Knowledge Based Systems*, 76(1):30–41, 2015.

Generating corrupted data sources for the evaluation of matching systems

Fiona McNeill[1][0000−0001−7873−5187], Diana Bental[1][0000−0003−3834−416X], Alasdair J G Gray[1][0000−0002−5711−4872], Sabina Jedrzejczyk[1], and Ahmad Alsadeeqi[1]

Heriot-Watt University, Edinburgh, Scotland
{f.mcneill, d.bental,a.j.g.gray,sj22,aa1262}@hw.ac.uk

Abstract. One of the most difficult aspects of developing matching systems – whether for matching ontologies or for other types of mismatched data – is evaluation. The accuracy of matchers are usually evaluated by measuring the results produced by the systems against reference sets, but gold-standard reference sets are expensive and difficult to create. In this paper we introduce crptr, which generates multiple variations of different sorts of dataset, where the degree of variation is controlled, in order that they can be used to evaluate matchers in different context.

Keywords: Matching · Evaluation · Data Corruption.

1 Introduction

One of the central problems of data matching is the issue of evaluation: when a system returns a set of matches, how are we to determine whether they are correct or not? How exactly do we define what a correct match is, and how do we determine whether the proposed matches fall into that category? If we have a range of different options, how do we determine which is the 'best' match?

In this paper we describe the use of the crptr system to create evaluation datasets for matching. crptr was developed to simulate data quality issues for test datasets used for record linkage evaluation. It can create multiple similar datasets with varying amounts of variation controlled by input settings, and provides a clear mapping back to the original dataset. This creates training and evaluation sets for matchers to run against. We have extended the crptr system to deal with structure in a context where we want to corrupt data sources in order to evaluate the semantic rewriting of queries to unknown data sources.

In Section 2 we describe the crptr system and its original application domain. Section 3 then details how we extended crptr to address corruption of other data sets and of queries. We discuss issues around evaluation in Section 4 and touch on related work in Section 5 before concluding the paper in Section 6.

2 The crptr system

Synthetically generated data is a common approach for evaluating and testing data analysis and mining approaches [9]. However, the use of synthetically generated data fails to capture the messiness of real world data, i.e., they omit data quality issues [5]. To overcome this we developed crptr: a data corruption application that injects data errors and variations based on user requirements. crptr allows the user to control data quality in the generated dataset by simulating and injecting data corruptions into any dataset using known (non-random) methods that mimic real-world data quality issues (errors and variations). Applying these corruptions on a synthetic dataset enables the control of data quality, which makes the synthetic data more realistic and usable for evaluations. crptr contains many corruption methods that mimic commonly found data quality issues, e.g., typing errors, alternative spellings, and missing or swapped attributes, that can be used to simulate different corruption scenarios based on the experiment or project requirements.

crptr works by using a *corruption profile* that controls which methods are used and how much. The idea is that the profile attempts to capture the data quality characteristics of the dataset being modelled. The corruption profile consist of many factors that define the way data need to be corrupted such as the total number of records that need to be corrupted and the corruption methods required to be applied on the data. By controlling the factors of the corruption profile, the user can configure crptr to mimic the data quality characteristics that fit the purpose of the research.

3 Application of crptr to Query Rewriting

The CHAIn system (Combining Heterogenous Agencies' Information) [7] has been designed to support users to successfully query data from a wide range of different data sources, even when these data sources are not known in advance (e.g., data sources of new collaborators). It is primarily aimed at supporting decision makers during crisis response, but is applicable in many domains. Any queries pre-written to extract required information are likely to fail (i.e., not return any data) on these unknown or updated data sources because the queries were not written according to the structure and terminology of the target data source. However, the data sources may well have relevant information that is closely related to the query. CHAIn extracts data from the target source that approximately matches the query (i.e., exceeds a given threshold) and uses this data to rewrite the query so that it succeeds on the datasource. It returns (potentially) multiple rewritten queries, the mappings used to generate them, the data they retrieved and a similarity score $\in [0, 1]$, ranked in order of similarity.

Evaluation in this context therefore means determining whether the scores returned are a reasonable reflection of the distance between the original query and the rewritten query, and hence whether the ranked list is a reasonable ordering of the likely relevance of the responses to what the decision maker actually

wants to know. In this context, the matching is done between the schema of the query and the schema of the target datasource[1]. In order to mimic the process of rewriting a query designed for one data source to succeed on a different data source, we create a query based on the data in a particular data source (i.e., so that it would be able to successfully query that data source) and then introduce corruption reflecting naturally-occurring differences. We can either keep the query fixed and corrupt the data source in multiple ways, or keep the data source fixed and corrupt the query. In practice, we focused on corrupting datasources and then generating corrupted queries from these corrupted datasources - firstly, because it created a more generic process that was able to corrupt both datasources and queries; secondly, because it allows us to more easily focus on the part of the query that is relevant in this context, which is the terminology referring to the target datasource.

We therefore needed to extend the functionality of crptr in two ways. (*i*) We need to consider the domain in which this matching is occurring to determine how terms should be corrupted; (*ii*) Because there is a structural element to schema, we need to consider how this could be corrupted and extend the system to perform this.

In terms of the first requirement, some of the corruptions methods in crptr (e.g., those focusing on spelling errors) are not relevant, whilst others such as abbreviations, need to be adapted, as some kinds of abbreviations (e.g., of first names) are unlikely to occur in our data sources. We need to determine what kinds of mismatches are likely to occur in our domain, and determine what sources we can automatically extract them from. CHAIn is designed to be domain independent, and when addressing the problem of matching different (but similar) data sources in the general case, we need a domain-independent lexical resource to suggest the kinds of synonyms, hyponyms, hypernyms and meronyms that different creators of data sources in a similar domain may naturally use. We therefore turned to WordNet [8], a generic and widely used lexical resource, to allow us to do term corruption. WordNet does provide some information about abbreviations and acronyms which we are able to use in our matching, although additional resources that provide more relevant corruptions in this area would improve performance (but are hard to find).

In terms of the second requirement, we needed to make sure any potential structural change in the schema of a CSV file was considered. This is structurally simple, consisting of columns which are named and ordered, and thus structural changes are restricted to reorganisation (addition, deletion and reordering) of the columns. For SPARQL queries in general there are, of course, many more structural elements (e.g, the potential presence of SPARQL commands such as aggregate functions), and a complete list of potential structural mismatches would be more complicated. As we are only concerned with the terms in the query which correspond with those of the expected data source, we can ignore all of the additional SPARQL structure, stripping out the relevant terms and reinserting the new terms after matching.

[1] Matching at the data level is required when queries are partially instantiated.

4 Evaluation of crptr for different data formats

The quality of the crptr output depends on whether the corrupted data sources it produces are a reasonable facsimile of different but related data sources that would naturally be found. If this is the case then we can infer that the performance of a matching system when matching different data sources created by crptr is a good indication of the matchers performance in real-world settings, and that therefore crptr is a useful matching evaluation tool.

This depends on two things: (*i*) are the terms in the look up table a good approximation of terms that could be used interchangable or in a similar way: is it modelling genuine *semantic* and *syntactic* mismatches?; (*ii*) are the structural mismatches introduced through the corruption process a good approximation of how similar data sources may differ? The first is highly domain dependent. We use WordNet, which is a very widely used lexical resource. It is also likely to be of benefit to also use domain-specific ontologies and lexicographies for each particular domain; however, these are hard to find and often of questionable quality, so this kind of domain-specific corruption may be hard to perform. Matching in such domains is also more efficient for the same reasons. The second aspect is domain independent but format specific. For each format the system is extended to, an analysis of what structural mismatches are possible is necessary in order to demonstrate that the corruptions produced are plausible and thorough.

5 Related work

To the best of our knowledge, a system to generate reference sets (records, queries, RDF data sources, ontologies, etc) in order to evaluate matching in these domains is unique.

Since reference ontologies are expensive to generate and often not available, [6], automatically generated test sets have been used to evaluate ontology matching since the Benchmark Test Set was developed for the Ontology Alignment Evaluation Initiative in 2004 and updated in 2016 [3]. Several other generators were inspired by this, including Swing [4]. These tend to focus on OWL ontologies and are less broadly applicable than crptr. The range of methods they use are in some cases more sophisticated than our techniques, and in domains for which they are relevant, crptr could be improved by incorporated such approaches.

Aside from ontology matching, there is existing work on generating synthetic datasets with structural variations for relational and RDF data for use in benchmarking. The Linked Data Benchmark Council [2] has supported the development of configurable and scalable synthetic RDF datasets with similar irregularities to real data, including structural irregularities, specifically in the domains of social networks and semantic publishing. Existing work on generating structural variatons in RDF data (e.g. [2]) is intended to test the functionality and scalability of searches and the maintenance of RDF datasets. STBenchmark [1] generates test cases for schema mapping systems, taking an original dataset and applying structural and term variations. This is used to create benchmark

data for hand-mapping systems rather than for automated matching or querying. Our work could be extended with similar strategies to these to experiment with greater structural variations.

6 Conclusions

In this paper we have discussed using the crptr system for generating multiple similar datasets for evaluating matchers within different domains. We briefly described how crptr was developed to focus on records and then extended to deal with queries based on CSV files, and could be extended to deal with other kinds of data sources. We discussed what evaluation of these corruption systems means in different contexts.

References

1. Alexe, B., Tan, W.C., Velegrakis, Y.: Stbenchmark: towards a benchmark for mapping systems. Proceedings of the VLDB Endowment **1**(1), 230–244 (2008)
2. Angles, R., Boncz, P., Larriba-Pey, J., Fundulaki, I., Neumann, T., Erling, O., Neubauer, P., Martinez-Bazan, N., Kotsev, V., Toma, I.: The linked data benchmark council: a graph and rdf industry benchmarking effort. ACM SIGMOD Record **43**(1), 27–31 (2014)
3. Euzenat, J., Rooiu, M.E., Trojahn, C.: Ontology matching benchmarks: Generation, stability, and discriminability. Journal of Web Semantics **21**, 30 – 48 (2013). https://doi.org/https://doi.org/10.1016/j.websem.2013.05.002, http://www.sciencedirect.com/science/article/pii/S1570826813000188, special Issue on Evaluation of Semantic Technologies
4. Ferrara, A., Montanelli, S., Noessner, J., Stuckenschmidt, H.: Benchmarking matching applications on the semantic web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) The Semanic Web: Research and Applications. pp. 108–122. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
5. Hernández, M.A., Stolfo, S.J.: Real-world data is dirty: Data cleansing and the merge/purge problem. Data mining and knowledge discovery **2**(1), 9–37 (1998)
6. Ivanova, V., Bach, B., Pietriga, E., Lambrix, P.: Alignment cubes: Towards interactive visual exploration and evaluation of multiple ontology alignments. In: d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudré-Mauroux, P., Sequeda, J., Lange, C., Heflin, J. (eds.) The Semantic Web – ISWC 2017. pp. 400–417. Springer International Publishing, Cham (2017)
7. McNeill, F., Gkaniatsou, A., Bundy, A.: Dynamic data sharing from large data sources. In: Proceedings of 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014) (2014)
8. Miller, G.A.: Wordnet: A lexical database for English. Commun. ACM **38**(11), 39–41 (Nov 1995). https://doi.org/10.1145/219717.219748, http://doi.acm.org/10.1145/219717.219748
9. Tran, K.N., Vatsalan, D., Christen, P.: Geco: an online personal data generator and corruptor. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2473–2476. ACM (2013)

Results of the Ontology Alignment Evaluation Initiative 2019*

Alsayed Algergawy[1], Daniel Faria[2], Alfio Ferrara[3], Irini Fundulaki[4], Ian Harrow[5], Sven Hertling[6], Ernesto Jiménez-Ruiz[7,8], Naouel Karam[9], Abderrahmane Khiat[10], Patrick Lambrix[11], Huanyu Li[11], Stefano Montanelli[3], Heiko Paulheim[6], Catia Pesquita[12], Tzanina Saveta[4], Pavel Shvaiko[13], Andrea Splendiani[5], Elodie Thiéblin[14], Cássia Trojahn[14], Jana Vataščinová[15], Ondřej Zamazal[15], and Lu Zhou[16]

[1] Friedrich Schiller University Jena, Germany
alsayed.algergawy@uni-jena.de
[2] BioData.pt, INESC-ID, Lisbon, Portugal
dfaria@inesc-id.pt
[3] Università degli studi di Milano, Italy
{alfio.ferrara,stefano.montanelli}@unimi.it
[4] Institute of Computer Science-FORTH, Heraklion, Greece
{jsaveta,fundul}@ics.forth.gr
[5] Pistoia Alliance Inc., USA
{ian.harrow,andrea.splendiani}@pistoiaalliance.org
[6] University of Mannheim, Germany
{sven,heiko}@informatik.uni-mannheim.de
[7] City, University of London, UK
ernesto.jimenez-ruiz@city.ac.uk
[8] Department of Informatics, University of Oslo, Norway
ernestoj@ifi.uio.no
[9] Fraunhofer FOKUS, Berlin, Germany
naouel.karam@fokus.fraunhofer.de
[10] Fraunhofer IAIS, Sankt Augustin, Bonn, Germany
abderrahmane.khiat@iais.fraunhofer.de
[11] Linköping University & Swedish e-Science Research Center, Linköping, Sweden
{patrick.lambrix,huanyu.li}@liu.se
[12] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
cpesquita@di.fc.ul.pt
[13] TasLab, Trentino Digitale SpA, Trento, Italy
pavel.shvaiko@tndigit.it
[14] IRIT & Université Toulouse II, Toulouse, France
{cassia.trojahn,elodie.thieblin}@irit.fr
[15] University of Economics, Prague, Czech Republic
{jana.vatascinova,ondrej.zamazal}@vse.cz
[16] Data Semantics (DaSe) Laboratory, Kansas State University, USA
luzhou@ksu.edu

Abstract. The Ontology Alignment Evaluation Initiative (OAEI) aims at comparing ontology matching systems on precisely defined test cases. These test cases can be based on ontologies of different levels of complexity (from simple thesauri to expressive OWL ontologies) and use different evaluation modalities (e.g., blind evaluation, open evaluation, or consensus). The OAEI 2019 campaign offered 11 tracks with 29 test cases, and was attended by 20 participants. This paper is an overall presentation of that campaign.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative, which organizes the evaluation of an increasing number of ontology matching systems [21, 23]. The main goal of the OAEI is to compare systems and algorithms openly and on the same basis, in order to allow anyone to draw conclusions about the best matching strategies. Furthermore, our ambition is that, from such evaluations, developers can improve their systems.

Two first events were organized in 2004: *(i)* the Information Interpretation and Integration Conference (I3CON) held at the NIST Performance Metrics for Intelligent Systems (PerMIS) workshop and *(ii)* the Ontology Alignment Contest held at the Evaluation of Ontology-based Tools (EON) workshop of the annual International Semantic Web Conference (ISWC) [48]. Then, a unique OAEI campaign occurred in 2005 at the workshop on Integrating Ontologies held in conjunction with the International Conference on Knowledge Capture (K-Cap) [5]. From 2006 until the present, the OAEI campaigns were held at the Ontology Matching workshop, collocated with ISWC [4, 1–3, 7, 8, 10, 13, 17–20, 22], which this year took place in Auckland, New Zealand[2].

Since 2011, we have been using an environment for automatically processing evaluations (§2.1) which was developed within the SEALS (Semantic Evaluation At Large Scale) project[3]. SEALS provided a software infrastructure for automatically executing evaluations and evaluation campaigns for typical semantic web tools, including ontology matching. Since OAEI 2017, a novel evaluation environment called HOBBIT (§2.1) was adopted for the HOBBIT Link Discovery track, and later extended to enable the evaluation of other tracks. Some tracks are run exclusively through SEALS and others through HOBBIT, but several allow participants to choose the platform they prefer.

This paper synthesizes the 2019 evaluation campaign and introduces the results provided in the papers of the participants. The remainder of the paper is organized as follows: in §2, we present the overall evaluation methodology; in §3 we present the tracks and datasets; in §4 we present and discuss the results; and finally, §5 discusses the lessons learned.

[1] http://oaei.ontologymatching.org
[2] http://om2019.ontologymatching.org
[3] http://www.seals-project.eu

2 Methodology

2.1 Evaluation platforms

The OAEI evaluation was carried out in one of two alternative platforms: the SEALS client or the HOBBIT platform. Both have the goal of ensuring reproducibility and comparability of the results across matching systems.

The **SEALS client** was developed in 2011. It is a Java-based command line interface for ontology matching evaluation, which requires system developers to implement a simple interface and to wrap their tools in a predefined way including all required libraries and resources. A tutorial for tool wrapping is provided to the participants, describing how to wrap a tool and how to run a full evaluation locally.

The **HOBBIT platform**[4] was introduced in 2017. It is a web interface for linked data and ontology matching evaluation, which requires systems to be wrapped inside docker containers and includes a SystemAdapter class, then being uploaded into the HOBBIT platform [34].

Both platforms compute the standard evaluation metrics against the reference alignments: precision, recall and F-measure. In test cases where different evaluation modalities are required, evaluation was carried out *a posteriori*, using the alignments produced by the matching systems.

2.2 OAEI campaign phases

As in previous years, the OAEI 2019 campaign was divided into three phases: preparatory, execution, and evaluation.

In the **preparatory phase**, the test cases were provided to participants in an initial assessment period between June 15^{th} and July 15^{th}, 2019. The goal of this phase is to ensure that the test cases make sense to participants, and give them the opportunity to provide feedback to organizers on the test case as well as potentially report errors. At the end of this phase, the final test base was frozen and released.

During the ensuing **execution phase**, participants test and potentially develop their matching systems to automatically match the test cases. Participants can self-evaluate their results either by comparing their output with the reference alignments or by using either of the evaluation platforms. They can tune their systems with respect to the non-blind evaluation as long as they respect the rules of the OAEI. Participants were required to register their systems and make a preliminary evaluation by July 31^{st}. The execution phase was terminated on September 30^{th}, 2019, at which date participants had to submit the (near) final versions of their systems (SEALS-wrapped and/or HOBBIT-wrapped).

During the **evaluation phase**, systems were evaluated by all track organizers. In case minor problems were found during the initial stages of this phase, they were reported to the developers, who were given the opportunity to fix and resubmit their systems. Initial results were provided directly to the participants, whereas final results for most tracks were published on the respective OAEI web pages by October 14^{th}, 2019.

[4] `https://project-hobbit.eu/outcomes/hobbit-platform/`

3 Tracks and test cases

This year's OAEI campaign consisted of 11 tracks gathering 29 test cases, all of which were based on OWL ontologies. They can be grouped into:

- Schema matching tracks, which have as objective matching ontology classes and/or properties.
- Instance Matching tracks, which have as objective matching ontology instances.
- Instance and Schema Matching tracks, which involve both of the above.
- Complex Matching tracks, which have as objective finding complex correspondences between ontology entities.
- Interactive tracks, which simulate user interaction to enable the benchmarking of interactive matching algorithms.

The tracks are summarized in Table 1.

Table 1. Characteristics of the OAEI tracks.

Track	Test Cases (Tasks)	Relations	Confidence	Evaluation	Languages	Platform
Schema Matching						
Anatomy	1	=	[0 1]	open	EN	SEALS
Biodiversity & Ecology	2	=	[0 1]	open	EN	SEALS
Conference	1 (21)	=, <=	[0 1]	open+blind	EN	SEALS
Disease & Phenotype	2	=, <=	[0 1]	open+blind	EN	SEALS
Large Biomedical ontologies	6	=	[0 1]	open	EN	both
Multifarm	2 (2445)	=	[0 1]	open+blind	AR, CZ, CN, DE, EN, ES, FR, IT, NL, RU, PT	SEALS
Instance Matching						
Link Discovery	2 (9)	=	[0 1]	open	EN	HOBBIT
SPIMBENCH	2	=	[0 1]	open+blind	EN	HOBBIT
Instance and Schema Matching						
Knowledge Graph	5	=	[0 1]	open	EN	SEALS
Interactive Matching						
Interactive	2 (22)	=, <=	[0 1]	open	EN	SEALS
Complex Matching						
Complex	4	=, <=, >=	[0 1]	open+blind	EN, ES	SEALS

Open evaluation is made with already published reference alignments and blind evaluation is made by organizers, either from reference alignments unknown to the participants or manually.

3.1 Anatomy

The anatomy track comprises a single test case consisting of matching two fragments of biomedical ontologies which describe the human anatomy[5] (3304 classes) and the anatomy of the mouse[6] (2744 classes). The evaluation is based on a manually curated reference alignment. This dataset has been used since 2007 with some improvements over the years [15].

Systems are evaluated with the standard parameters of precision, recall, F-measure. Additionally, recall+ is computed by excluding trivial correspondences (i.e., correspondences that have the same normalized label). Alignments are also checked for coherence using the Pellet reasoner. The evaluation was carried out on a server with a 6 core CPU @ 3.46 GHz with 8GB allocated RAM, using the SEALS client. However, the evaluation parameters were computed *a posteriori*, after removing from the alignments produced by the systems, correspondences expressing relations other than equivalence, as well as trivial correspondences in the oboInOwl namespace (e.g., oboInOwl#Synonym = oboInOwl#Synonym). The results obtained with the SEALS client vary in some cases by 0.5% compared to the results presented below.

3.2 Biodiversity and Ecology

The second edition of biodiversity track features two test cases based on highly overlapping ontologies that are particularly useful for biodiversity and ecology research: matching Environment Ontology (ENVO) to Semantic Web for Earth and Environment Technology Ontology (SWEET), and matching Flora Phenotype Ontology (FLOPO) to Plant Trait Ontology (PTO). The track was motivated by two projects, namely GFBio[7] (The German Federation for Biological Data) and AquaDiva[8], which aim at providing semantically enriched data management solutions for data capture, annotation, indexing and search [35, 37]. Table 2 summarizes the versions and the sizes of the ontologies used in OAEI 2019. Compared to the first edition, the number of concepts of the ENVO and FOLPO ontologies has increased, which required the creation of new reference alignments for both tasks.

Table 2. Versions and number of classes of the Biodiversity and Ecology track ontologies.

Ontology	Version	Classes
ENVO	2019-03-18	8968
SWEET	2018-03-12	4543
FLOPO	2016-06-03	28965
PTO	2017-09-11	1504

[5] www.cancer.gov/cancertopics/cancerlibrary/terminologyresources
[6] http://www.informatics.jax.org/searches/AMA_form.shtml
[7] www.gfbio.org
[8] www.aquadiva.uni-jena.de

To this end, we updated the reference alignments for the two test cases following the same procedure as in the first edition. In particular, alignment files were produced through a hybrid approach consisting of (1) an updated consensus alignment based on matching systems output, then (2) manually validating a subset of unique mappings produced by each system (and adding them to the consensus if considered correct), and finally (3) adding a set of manually generated correspondences. The matching systems used to generate the consensus alignments were those participating in this track last year [4], namely: AML, Lily, LogMap family, POMAP and XMAP.

The evaluation was carried out on a Windows 10 (64-bit) desktop with an Intel Core i5-7500 CPU @ 3.40GHz x 4 with 15.7 Gb RAM allocated, using the SEALS client. Systems were evaluated using the standard metrics.

3.3 Conference

The conference track features a single test case that is a suite of 21 matching tasks corresponding to the pairwise combination of 7 moderately expressive ontologies describing the domain of organizing conferences. The dataset and its usage are described in [52].

The track uses several reference alignments for evaluation: the old (and not fully complete) manually curated open reference alignment, *ra1*; an extended, also manually curated version of this alignment, *ra2*; a version of the latter corrected to resolve violations of conservativity, *rar2*; and an uncertain version of *ra1* produced through crowd-sourcing, where the score of each correspondence is the fraction of people in the evaluation group that agree with the correspondence. The latter reference was used in two evaluation modalities: *discrete* and *continuous* evaluation. In the former, correspondences in the uncertain reference alignment with a score of at least 0.5 are treated as correct whereas those with lower score are treated as incorrect, and standard evaluation parameters are used to evaluated systems. In the latter, weighted precision, recall and F-measure values are computed by taking into consideration the actual scores of the uncertain reference, as well as the scores generated by the matching system. For the sharp reference alignments (*ra1*, *ra2* and *rar2*), the evaluation is based on the standard parameters, as well the $F_{0.5}$-measure and F_2-measure and on conservativity and consistency violations. Whereas F_1 is the harmonic mean of precision and recall where both receive equal weight, F_2 gives higher weight to recall than precision and $F_{0.5}$ gives higher weight to precision higher than recall.

Two baseline matchers are used to benchmark the systems: edna string edit distance matcher; and StringEquiv string equivalence matcher as in the anatomy test case.

The evaluation was carried out on a Windows 10 (64-bit) desktop with an Intel Core i7–8550U (1,8 GHz, TB 4 GHz) x 4 with 16 GB RAM allocated using the SEALS client. Systems were evaluated using the standard metrics.

3.4 Disease and Phenotype

The Disease and Phenotype is organized by the Pistoia Alliance Ontologies Mapping project team[9]. It comprises 2 test cases that involve 4 biomedical ontologies covering the disease and phenotype domains: Human Phenotype Ontology (HP) versus

[9] http://www.pistoiaalliance.org/projects/ontologies-mapping/

Mammalian Phenotype Ontology (MP) and Human Disease Ontology (DOID) versus Orphanet and Rare Diseases Ontology (ORDO). Currently, correspondences between these ontologies are mostly curated by bioinformatics and disease experts who would benefit from automation of their workflows supported by implementation of ontology matching algorithms. More details about the Pistoia Alliance Ontologies Mapping project and the OAEI evaluation are available in [25]. Table 3.4 summarizes the versions of the ontologies used in OAEI 2019.

Table 3. Disease and Phenotype ontology versions and sources.

Ontology	Version	Source
HP	2017-06-30	OBO Foundry
MP	2017-06-29	OBO Foundry
DOID	2017-06-13	OBO Foundry
ORDO	v2.4	ORPHADATA

The reference alignments used in this track are silver standard consensus alignments automatically built by merging/voting the outputs of the participating systems in 2016, 2017 and 2018 (with vote=3). Note that systems participating with different variants and in different years only contributed once in the voting, that is, the voting was done by family of systems/variants rather than by individual systems. The HP-MP silver standard thus produced contains 2232 correspondences, whereas the DOID-ORDO one contains 2808 correspondences.

Systems were evaluated using the standard parameters as well as the number of unsatisfiable classes computed using the OWL 2 reasoner HermiT [41]. The evaluation was carried out in a Ubuntu 18 Laptop with an Intel Core i5-6300HQ CPU @ 2.30GHz x 4 and allocating 15 Gb of RAM.

3.5 Large Biomedical Ontologies

The large biomedical ontologies (largebio) track aims at finding alignments between the large and semantically rich biomedical ontologies FMA, SNOMED-CT, and NCI, which contain 78,989, 306,591 and 66,724 classes, respectively. The track consists of six test cases corresponding to three matching problems (FMA-NCI, FMA-SNOMED and SNOMED-NCI) in two modalities: small overlapping fragments and whole ontologies (FMA and NCI) or large fragments (SNOMED-CT).

The reference alignments used in this track are derived directly from the UMLS Metathesaurus [6] as detailed in [32], then automatically repaired to ensure logical coherence. However, rather than use a standard repair procedure of removing problem causing correspondences, we set the relation of such correspondences to *"?"* (unknown). These "?" correspondences are neither considered positive nor negative when evaluating matching systems, but are simply ignored. This way, systems that do not perform alignment repair are not penalized for finding correspondences that (despite causing incoherences) may or may not be correct, and systems that do perform alignment repair are not penalized for removing such correspondences. To avoid any bias,

correspondences were considered problem causing if they were selected for removal by any of the three established repair algorithms: Alcomo [39], LogMap [31], or AML [43]. The reference alignments are summarized in Table 4.

Table 4. Number of correspondences in the reference alignments of the large biomedical ontologies tasks.

Reference alignment	"=" corresp.	"?" corresp.
FMA-NCI	2,686	338
FMA-SNOMED	6,026	2,982
SNOMED-NCI	17,210	1,634

The evaluation was carried out in a Ubuntu 18 Laptop with an Intel Core i5-6300HQ CPU @ 2.30GHz x 4 and allocating 15 Gb of RAM. Evaluation was based on the standard parameters (modified to account for the "?" relations) as well as the number of unsatisfiable classes and the ratio of unsatisfiable classes with respect to the size of the union of the input ontologies. Unsatisfiable classes were computed using the OWL 2 reasoner HermiT [41], or, in the cases in which HermiT could not cope with the input ontologies and the alignments (in less than 2 hours) a lower bound on the number of unsatisfiable classes (indicated by $\geq$) was computed using the OWL2 EL reasoner ELK [36].

3.6 Multifarm

The multifarm track [40] aims at evaluating the ability of matching systems to deal with ontologies in different natural languages. This dataset results from the translation of 7 ontologies from the conference track (cmt, conference, confOf, iasted, sigkdd, ekaw and edas) into 10 languages: Arabic (ar), Chinese (cn), Czech (cz), Dutch (nl), French (fr), German (de), Italian (it), Portuguese (pt), Russian (ru), and Spanish (es). The dataset is composed of 55 pairs of languages, with 49 matching tasks for each of them, taking into account the alignment direction (e.g. $\text{cmt}_{en} \rightarrow \text{edas}_{de}$ and $\text{cmt}_{de} \rightarrow \text{edas}_{en}$ are distinct matching tasks). While part of the dataset is openly available, all matching tasks involving the *edas* and *ekaw* ontologies (resulting in 55×24 matching tasks) are used for blind evaluation.

We consider two test cases: i) those tasks where two different ontologies (cmt→edas, for instance) have been translated into two different languages; and ii) those tasks where the same ontology (cmt→cmt) has been translated into two different languages. For the tasks of type ii), good results are not only related to the use of specific techniques for dealing with cross-lingual ontologies, but also on the ability to exploit the identical structure of the ontologies.

The reference alignments used in this track derive directly from the manually curated Conference *ra1* reference alignments. The systems have been executed on a Ubuntu Linux machine configured with 8GB of RAM running under a Intel Core CPU 2.00GHz x4 processors, using the SEALS client.

3.7 Link Discovery

The Link Discovery track features two test cases, Linking and Spatial, that deal with *link discovery* for spatial data represented as *trajectories* i.e., sequences of longitude, latitude pairs. The track is based on two datasets generated from TomTom[10] and Spaten [12].

The **Linking** test case aims at testing the performance of instance matching tools that implement mostly string-based approaches for identifying matching entities. It can be used not only by instance matching tools, but also by SPARQL engines that deal with query answering over geospatial data. The test case was based on SPIMBENCH [44], but since the ontologies used to represent trajectories are fairly simple and do not consider complex RDF or OWL schema constructs already supported by SPIMBENCH, only a subset of the transformations implemented by SPIMBENCH was used. The transformations implemented in the test case were (i) string-based with different (a) levels, (b) types of spatial object representations and (c) types of date representations, and (ii) schema-based, i.e., addition and deletion of ontology (schema) properties. These transformations were implemented in the TomTom dataset. In a nutshell, instance matching systems are expected to determine whether two traces with their points annotated with place names designate the same trajectory. In order to evaluate the systems a ground truth was built that contains the set of expected links where an instance s_1 in the source dataset is associated with an instance t_1 in the target dataset that has been generated as a modified description of s_1.

The **Spatial** test case aims at testing the performance of systems that deal with topological relations proposed in the state of the art DE-9IM (Dimensionally Extended nine-Intersection Model) model [47]. The benchmark generator behind this test case implements all topological relations of DE-9IM between trajectories in the two dimensional space. To the best of our knowledge such a generic benchmark, that takes as input trajectories and checks the performance of linking systems for spatial data does not exist. The focus for the design was (a) on the correct implementation of all the topological relations of the DE-9IM topological model and (b) on producing datasets large enough to stress the systems under test. The supported relations are: *E*quals, Disjoint, Touches, Contains/Within, Covers/CoveredBy, Intersects, Crosses, Overlaps. The test case comprises tasks for all the DE-9IM relations and for LineString/LineString and LineString/Polygon cases, for both TomTom and Spaten datasets, ranging from 200 to 2K instances. We did not exceed 64 KB per instance due to a limitation of the Silk system[11], in order to enable a fair comparison of the systems participating in this track.

The evaluation for both test cases was carried out using the HOBBIT platform.

3.8 SPIMBENCH

The **SPIMBENCH** track consists of matching instances that are found to refer to the same real-world entity corresponding to a creative work (that can be a news item,

[10] `https://www.tomtom.com/en_gr/`

[11] `https://github.com/silk-framework/silk/issues/57`

blog post or programme). The datasets were generated and transformed using SPIMBENCH [44] by altering a set of original linked data through value-based, structure-based, and semantics-aware transformations (simple combination of transformations). They share almost the same ontology (with some differences in property level, due to the structure-based transformations), which describes instances using 22 classes, 31 data properties, and 85 object properties. Participants are requested to produce a set of correspondences between the pairs of matching instances from the source and target datasets that are found to refer to the same real-world entity. An instance in the source dataset can have none or one matching counterpart in the target dataset. The SPIMBENCH task uses two sets of datasets[12] with different scales (i.e., number of instances to match):

- Sandbox (380 INSTANCES, 10000 TRIPLES). It contains two datasets called source (Tbox1) and target (Tbox2) as well as the set of expected correspondences (i.e., reference alignment).
- Mainbox (1800 CWs, 50000 TRIPLES). It contains two datasets called source (Tbox1) and target (Tbox2). This test case is blind, meaning that the reference alignment is not given to the participants.

In both cases, the goal is to discover the correspondences among the instances in the source dataset (Tbox1) and the instances in the target dataset (Tbox2).

The evaluation was carried out using the HOBBIT platform.

3.9 Knowledge Graph

The Knowledge Graph track was run for the second year. The task of the track is to match pairs of knowledge graphs, whose schema and instances have to be matched simultaneously. The individual knowledge graphs are created by running the DBpedia extraction framework on eight different Wikis from the Fandom Wiki hosting platform[13] in the course of the DBkWik project [27, 26]. They cover different topics (movies, games, comics and books) and three Knowledge Graph clusters shares the same domain e.g. star trek, as shown in Table 5.

The evaluation is based on reference correspondences at both schema and instance levels. While the schema level correspondences were created by experts, the instance correspondences were extracted from the wiki page itself. Due to the fact that not all inter wiki links on a page represent the same concept a few restrictions were made: 1) Only links in sections with a header containing "link" are used 2) all links are removed where the source page links to more than one concept in another wiki (ensures the alignments are functional) 3) multiple links which point to the same concept are also removed (ensures injectivity). Since we do not have a correspondence for each instance, class, and property in the graphs, this gold standard is only a *partial gold standard*.

The evaluation was executed on a virtual machine (VM) with 32GB of RAM and 16 vCPUs (2.4 GHz), with Debian 9 operating system and Openjdk version 1.8.0_212, using the SEALS client (version 7.0.5). We used the $-\circ$ option in SEALS to provide the

[12] Although the files are called Tbox1 and Tbox2, they actually contain a Tbox and an Abox.

[13] `https://www.wikia.com/`

Table 5. Characteristics of the Knowledge Graphs in the Knowledge Graph track, and the sources they were created from.

Source	Hub	Topic	#Instances	#Properties	#Classes
Star Wars Wiki	Movies	Entertainment	145,033	700	269
The Old Republic Wiki	Games	Gaming	4,180	368	101
Star Wars Galaxies Wiki	Games	Gaming	9,634	148	67
Marvel Database	Comics	Comics	210,996	139	186
Marvel Cinematic Universe	Movies	Entertainment	17,187	147	55
Memory Alpha	TV	Entertainment	45,828	325	181
Star Trek Expanded Universe	TV	Entertainment	13,426	202	283
Memory Beta	Books	Entertainment	51,323	423	240

two knowledge graphs which should be matched. We used local files rather than HTTP URLs to circumvent the overhead of downloading the knowledge graphs. We could not use the "-x" option of SEALS because the evaluation routine needed to be changed for two reasons: first, to differentiate between results for class, property, and instance correspondences, and second, to deal with the partial nature of the gold standard.

The alignments were evaluated based on precision, recall, and f-measure for classes, properties, and instances (each in isolation). The partial gold standard contained 1:1 correspondences and we further assume that in each knowledge graph, only one representation of the concept exists. This means that if we have a correspondence in our gold standard, we count a correspondence to a different concept as a false positive. The count of false negatives is only increased if we have a 1:1 correspondence and it is not found by a matcher. The whole source code for generating the evaluation results is also available[14].

As a baseline, we employed two simple string matching approaches. The source code for these matchers is publicly available[15].

3.10 Interactive Matching

The interactive matching track aims to assess the performance of semi-automated matching systems by simulating user interaction [42, 14, 38]. The evaluation thus focuses on how interaction with the user improves the matching results. Currently, this track does not evaluate the user experience or the user interfaces of the systems [29, 14].

The interactive matching track is based on the datasets from the Anatomy and Conference tracks, which have been previously described. It relies on the SEALS client's *Oracle* class to simulate user interactions. An interactive matching system can present a collection of correspondences simultaneously to the oracle, which will tell the system whether that correspondence is correct or not. If a system presents up to three correspondences together and each correspondence presented has a mapped entity (i.e., class

[14] `http://oaei.ontologymatching.org/2019/results/knowledgegraph/matching-eval-trackspecific.zip`

[15] `http://oaei.ontologymatching.org/2019/results/knowledgegraph/kgBaselineMatchers.zip`

or property) in common with at least one other correspondence presented, the oracle counts this as a single interaction, under the rationale that this corresponds to a scenario where a user is asked to choose between conflicting candidate correspondences. To simulate the possibility of user errors, the oracle can be set to reply with a given error probability (randomly, from a uniform distribution). We evaluated systems with four different error rates: 0.0 (perfect user), 0.1, 0.2, and 0.3.

In addition to the standard evaluation parameters, we also compute the number of requests made by the system, the total number of distinct correspondences asked, the number of positive and negative answers from the oracle, the performance of the system according to the oracle (to assess the impact of the oracle errors on the system) and finally, the performance of the oracle itself (to assess how erroneous it was).

The evaluation was carried out on a server with 3.46 GHz (6 cores) and 8GB RAM allocated to the matching systems. Each system was run ten times and the final result of a system for each error rate represents the average of these runs. For the Conference dataset with the ra1 alignment, precision and recall correspond to the micro-average over all ontology pairs, whereas the number of interactions is the total number of interactions for all the pairs.

3.11 Complex Matching

The complex matching track is meant to evaluate the matchers based on their ability to generate complex alignments. A complex alignment is composed of complex correspondences typically involving more than two ontology entities, such as o_1*:AcceptedPaper* $\equiv$ o_2*:Paper* $\sqcap$ o_2*:hasDecision.*o_2*:Acceptance*. Four datasets with their own evaluation process have been proposed [51].

The **complex conference** dataset is composed of three ontologies: cmt, conference and ekaw from the conference dataset. The reference alignment was created as a consensus between experts. In the evaluation process, the matchers can take the simple reference alignment *ra1* as input. The precision and recall measures are manually calculated over the complex equivalence correspondences only.

The **populated complex conference** is a populated version of the Conference dataset. 5 ontologies have been populated with more or less common instances resulting in 6 datasets (6 versions on the seals repository: v0, v20, v40, v60, v80 and v100). The alignments were evaluated based on Competency Questions for Alignment, i.e., basic queries that the alignment should be able to cover [49]. The queries are automatically rewritten using 2 systems: that from [50] which covers (1:n) correspondences with EDOAL expressions; and a system which compares the answers (sets of instances or sets of pairs of instances) of the source query and the source member of the correspondences and which outputs the target member if both sets are identical. The best rewritten query scores are kept. A precision score is given by comparing the instances described by the source and target members of the correspondences.

The **Hydrography** dataset consists of matching four different source ontologies (hydro3, hydrOntology-translated, hydrOntology-native, and cree) to a single target ontology (SWO) [9]. The evaluation process is based on three subtasks: given an entity from the source ontology, identify all related entities in the source and target ontology; given an entity in the source ontology and the set of related entities, identify the logical

relation that holds between them; identify the full complex correspondences. The three subtasks were evaluated based on relaxed precision and recall [16].

The **GeoLink** dataset derives from the homonymous project, funded under the U.S. National Science Foundation's EarthCube initiative. It is composed of two ontologies: the GeoLink Base Ontology (GBO) and the GeoLink Modular Ontology (GMO). The GeoLink project is a real-world use case of ontologies, and the instance data is also available and populated into the benchmark. The alignment between the two ontologies was developed in consultation with domain experts from several geoscience research institutions. More detailed information on this benchmark can be found in [54, 55]. Evaluation was done in the same way as with the Hydrography dataset. The evaluation platform was a MacBook Pro with a 2.5 GHz Intel Core i7 processor and 16 GB of 1600 MHz DDR3 RAM running mac OS Yosemite version 10.10.5.

The **Taxon** dataset is composed of four knowledge bases containing knowledge about plant taxonomy: AgronomicTaxon, AGROVOC, TAXREF-LD and DBpedia. The evaluation is two-fold: first, the precision of the output alignment is manually assessed; then, a set of source queries are rewritten using the output alignment. The rewritten target query is then manually classified as correct or incorrect. A source query is considered successfully rewritten if at least one of the target queries is semantically equivalent to it. The proportion of source queries successfully rewritten is then calculated (QWR in the results table). The evaluation over this dataset is open to all matching systems (simple or complex) but some queries can not be rewritten without complex correspondences. The evaluation was performed with an Ubuntu 16.04 machine configured with 16GB of RAM running under a i7-4790K CPU 4.00GHz x 8 processors.

4 Results and Discussion

4.1 Participation

Following an initial period of growth, the number of OAEI participants has remained approximately constant since 2012, which is slightly over 20. This year we count with 20 participating systems. Table 6 lists the participants and the tracks in which they competed. Some matching systems participated with different variants (AML, LogMap) whereas others were evaluated with different configurations, as requested by developers (see test case sections for details).

A number of participating systems use external sources of background knowledge, which are especially critical in matching ontologies in the biomedical domain. LogMap-Bio uses BioPortal as mediating ontology provider, that is, it retrieves from BioPortal the most suitable top-10 ontologies for each matching task. LogMap uses normalizations and spelling variants from the general (biomedical) purpose SPECIALIST Lexicon. AML has three sources of background knowledge which can be used as mediators between the input ontologies: the Uber Anatomy Ontology (Uberon), the Human Disease Ontology (DOID) and the Medical Subject Headings (MeSH). XMAP and Lily use a dictionary of synonyms (pre)extracted from the UMLS Metathesaurus. In addition Lily also uses a dictionary of synonyms (pre)extracted from BioPortal.

Table 6. Participants and the status of their submissions.

System	AGM	ALIN	AML	AMLC	AROA	CANARD	DOME	EVOCROS	FCAMap-KG	FTRLIM	Lily	LogMap	LogMap-Bio	LogMapLt	OntMat1	POMAP++	RADON	SANOM	Silk	WktMtchr	Total=20
Confidence	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
anatomy	●	●	●	○	○	○	●	○	●	○	●	●	●	●	○	●	○	●	○	●	12
conference	○	●	●	○	○	○	●	○	○	○	●	●	○	●	●	○	○	●	○	●	9
multifarm	○	○	●	○	○	○	○	○	○	○	◐	●	○	○	○	○	○	○	○	●	4
complex	○	○	○	◐	◐	◐	○	○	○	○	○	○	○	○	○	○	○	○	○	○	3
interactive	○	●	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	3
largebio	●	○	●	○	○	○	●	○	●	○	○	●	●	●	○	●	○	●	○	●	10
phenotype	○	○	●	○	○	○	●	○	●	○	○	●	●	●	○	●	○	○	○	●	8
biodiv	○	○	●	○	○	○	●	○	●	○	○	●	●	●	○	●	○	○	○	○	7
spimbench	○	○	●	○	○	○	○	○	○	●	●	●	○	○	○	○	●	○	●	○	6
link discovery	○	○	●	○	○	○	○	○	○	●	●	○	○	○	○	○	●	○	●	○	6
knowledge graph	●	○	●	○	○	○	●	○	●	○	○	●	●	●	○	●	○	○	○	●	9
total	3	3	10	1	1	1	6	0	5	2	5	10	5	6	1	5	2	3	2	5	77

Confidence pertains to the confidence scores returned by the system, with ✓ indicating that they are non-boolean; ○ indicates that the system did not participate in the track; ● indicates that it participated fully in the track; and ◐ indicates that it participated in or completed only part of the tasks of the track.

4.2 Anatomy

The results for the Anatomy track are shown in Table 7. Of the 12 systems participating in the Anatomy track, 10 achieved an F-measure higher than the StringEquiv baseline. Two systems were first time participants (Wiktionary and AGM). Long-term participating systems showed few changes in comparison with previous years with respect to alignment quality (precision, recall, F-measure, and recall+), size and run time. The exceptions were LogMapBio which increased in both recall+ (from 0.756 to 0.801) and alignment size (by 57 correspondences) since last year, and ALIN that increased in F-measure (from 0.758 to 0.813) and recall+ (from 0.0 to 0.365), as well as had a substantial increase of 158 correspondences since last year.

In terms of run time, 5 out of 12 systems computed an alignment in less than 100 seconds, a ratio which is similar to 2018 (6 out of 14). LogMapLite remains the system with the shortest runtime. Regarding quality, AML remains the system with the highest F-measure (0.943) and recall+ (0.832), but 3 other systems obtained an F-measure above 0.88 (LogMapBio, POMap++, and LogMap) which is at least as good as the best systems in OAEI 2007-2010. Like in previous years, there is no significant correlation between the quality of the generated alignment and the run time. Four systems produced coherent alignments.

Table 7. Anatomy results, ordered by F-measure. Runtime is measured in seconds; "size" is the number of correspondences in the generated alignment.

System	Runtime	Size	Precision	F-measure	Recall	Recall+	Coherent
AML	76	1493	0.95	0.943	0.936	0.832	√
LogMapBio	1718	1607	0.872	0.898	0.925	0.801	√
POMAP++	345	1446	0.919	0.897	0.877	0.695	-
LogMap	28	1397	0.918	0.88	0.846	0.593	√
SANOM	516	-	0.888	0.865	0.844	0.632	-
Lily	281	1381	0.873	0.833	0.796	0.52	-
Wiktionary	104	1144	0.968	0.832	0.73	0.288	-
LogMapLite	19	1147	0.962	0.828	0.728	0.288	-
ALIN	5115	1086	0.974	0.813	0.698	0.365	√
FCAMap-KG	25	960	0.996	0.772	0.631	0.042	-
StringEquiv	-	946	0.997	0.766	0.622	0.000	-
DOME	23	936	0.996	0.76	0.615	0.007	-
AGM	628	1942	0.152	0.171	0.195	0.154	-

4.3 Biodiversity and Ecology

Five of the systems participating this year had participated in this track in OAEI 2018: AML, LogMap family systems (LogMap, LogMapBio and LogMapLT) and POMAP. Three were new participants: DOME, FCAMapKG and LogMapKG. The newcomers DOME, FCAMapKG did not register explicitly to this track but could cope with at least one task so we did include their results.

We observed a slight increase in the number of systems (8 systems) that succeeded to generate alignments for the FLOPO-PTO task in comparison to previous year (7 systems). However, we witnessed a slight decrease in the number of systems (6 systems) that succeeded to generate alignments for the test ENVO-SWEET in comparison to previous year (7 systems). Lily did not manage to generate mappings for both tasks and LogMapBio did not manage to generated mappings for the ENVO-SWEET task.

As in the previous edition, we used precision, recall and F-measure to evaluate the performance of the participating systems. This year we included the execution times. The results for the Biodiversity and Ecology track are shown in Table 8.

Overall, the results of the participating systems have decreased in terms of F-measure for both tasks compared to last year. In terms of run time, most of the systems (except POMAP) computed an alignment in less than 100 seconds.

For the FLOPO-PTO task, AML and LogMapKG achieved the highest F-measure (0.78), with a slight difference in favor of AML. However, AML showed a remarkable decrease in terms of precision (from 0.88 to 0.76) and F-measure (from 0.86 to 0.78) compared to last year. LogMap also showed a slight decrease in terms of F-measure (from 0.80 to 0.78). The DOME system (newcomer) achieved the highest precision (0.99) with quite a good F-measure (0.739).

Regarding the ENVO-SWEET task, AML ranked first in terms of F-measure (0.80), followed by POMAP (0.69), FCAMapKG (0.63) and LogMapKG (0.63). As last year AML showed a very high recall and significant larger alignment than the other top

Table 8. Results for the Biodiversity & Ecology track.

System	Time (s)	Size	Precision	Recall	F-measure
FLOPO-PTO task					
AML	42	511	0.766	0.811	0.788
DOME	8.22	141	0.993	0.588	0.739
FCAMapKG	7.2	171	0.836	0.601	0.699
LogMap	14.4	235	0.791	0.782	0.768
LogMapBio	480.6	239	0.778	0.782	0.780
LogMapKG	13.2	235	0.791	0.782	0.786
LogMapLite	6.18	151	0.947	0.601	0.735
POMap	311	261	0.651	0.714	0.681
ENVO-SWEET task					
AML	3	925	0.733	0.899	0.808
FCAMapKG	7.8	422	0.803	0.518	0.630
LogMap	26.9	443	0.772	0.523	0.624
LogMapKG	7.98	422	0.803	0.518	0.630
LogMapLite	13.8	617	0.648	0.612	0.629
POMap	223	673	0.684	0.703	0.693

systems, but a comparably lower precision and a slight decrease in terms of F-measure (from 0.84 to 0.80). POMAP ranked second this year with a remarkable decrease in terms of precision (from 0.83 to 0.68) and F-measure (from 0.78 to 0.69). FCAMapKG and LogMapKG showed the highest results in terms of precision (0.80).

AML generated a significantly large number of mappings (much bigger than the size of the reference alignments for both tasks), those alignments were mostly subsumption mappings. In order to evaluate the precision in a more significant manner, we had to calculate an approximation by assessing manually a subset of mappings not present in the reference alignment (around a 100 for each task).

Overall, in this second evaluation, the results obtained from participating systems remained similar with a slight decrease in terms of F-measure compared to last year. It is worth noting that most of the participating systems, and all of the most successful ones use external resources as background knowledge.

4.4 Conference

The conference evaluation results using the sharp reference alignment *rar2* are shown in Table 9. For the sake of brevity, only results with this reference alignment and considering both classes and properties are shown. For more detailed evaluation results, please check conference track's web page.

With regard to two baselines we can group tools according to matcher's position: four matching systems outperformed both baselines (SANOM, AML, LogMap and Wiktionary); two performed the same as the edna baseline (DOME and LogMapLt); one performed slightly worse than this baseline (ALIN); and two (Lily and ONTMAT1) performed worse than both baselines. Three matchers (ONTMAT1, ALIN and Lily) do

Table 9. The highest average $F_{[0.5|1|2]}$-measure and their corresponding precision and recall for each matcher with its F_1-optimal threshold (ordered by F_1-measure). Inc.Align. means number of incoherent alignments. Conser.V. means total number of all conservativity principle violations. Consist.V. means total number of all consistency principle violations.

System	Prec.	$F_{0.5}$-m.	F_1-m.	F_2-m.	Rec.	Inc.Align.	Conser.V.	Consist.V.
SANOM	0.72	0.71	0.7	0.69	0.68	9	103	92
AML	0.78	0.74	0.69	0.65	0.62	0	39	0
LogMap	0.77	0.72	0.66	0.6	0.57	0	25	0
Wiktionary	0.65	0.62	0.58	0.54	0.52	7	133	27
DOME	0.73	0.65	0.56	0.5	0.46	3	105	10
edna	0.74	0.66	0.56	0.49	0.45			
LogMapLt	0.68	0.62	0.56	0.5	0.47	3	97	18
ALIN	0.81	0.68	0.55	0.46	0.42	0	2	0
StringEquiv	0.76	0.65	0.53	0.45	0.41			
Lily	0.54	0.53	0.52	0.51	0.5	9	140	124
ONTMAT1	0.77	0.64	0.52	0.43	0.39	1	71	37

not match properties at all. Naturally, this has a negative effect on their overall performance.

The performance of all matching systems regarding their precision, recall and F_1-measure is plotted in Figure 1. Systems are represented as squares or triangles, whereas the baselines are represented as circles.

With respect to logical coherence [45, 46], only three tools (ALIN, AML and LogMap) have no consistency principle violation (the same tools as last year). This year all tools have some conservativity principle violations as the last year). We should note that these conservativity principle violations can be "false positives" since the entailment in the aligned ontology can be correct although it was not derivable in the single input ontologies.

This year we additionally analyzed the False Positives, i.e. correspondences discovered by the tools which were evaluated as incorrect. The list of the False Positives is available on the conference track's web page. We looked at the reasons why a correspondence was incorrect or why it was discovered from a general point of view, and defined 3 reasons why alignments are incorrect and 5 reasons why they could have been chosen. Looking at the results, it can be said that when the reason a correspondence was discovered was the same name, all or at least most tools generated the correspondence. False Positives not discovered based on the same name or synonyms were produced by Lily, ONTMAT1 and SANOM. SANOM was the only tool which produced these correspondences based on similar strings. In three cases, a class was matched with a property by DOME (1x), LogMapLt (1x) and Wiktionary (3x).

The Conference evaluation results using the uncertain reference alignments are presented in Table 10.

Out of the 9 alignment systems, five (ALIN, DOME, LogMapLt, ONTMAT1, SANOM) use 1.0 as the confidence value for all matches they identify. The remaining

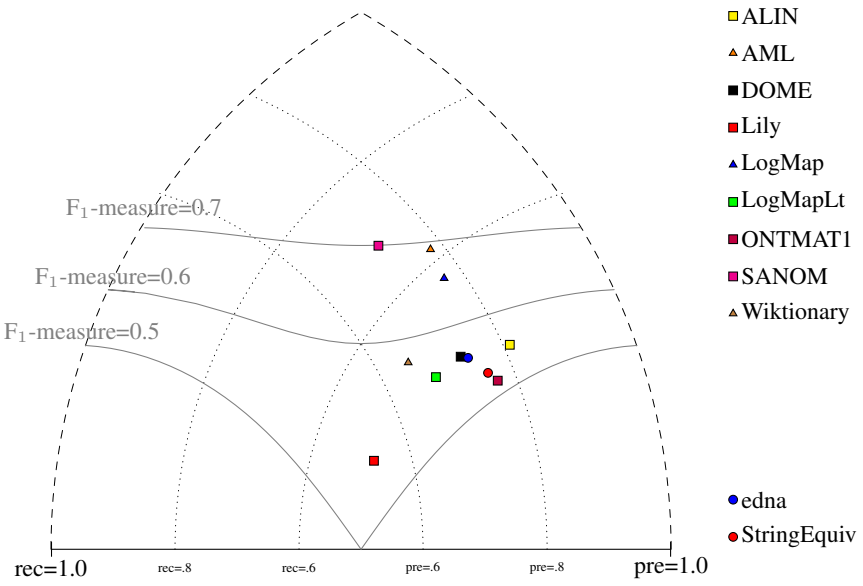


Fig. 1. Precision/recall triangular graph for the conference test case. Dotted lines depict level of precision/recall while values of F_1-measure are depicted by areas bordered by corresponding lines F_1-measure=0.[5|6|7].

Table 10. F-measure, precision, and recall of matchers when evaluated using the sharp (*ra1*), discrete uncertain and continuous uncertain metrics. Sorted according to F_1-m. in continuous.

System	Sharp			Discrete			Continuous		
	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.	Prec.	F_1-m.	Rec.
ALIN	0.87	0.58	0.44	0.87	0.68	0.56	0.87	0.69	0.57
AML	0.84	0.74	0.66	0.79	0.78	0.77	0.80	0.77	0.74
DOME	0.78	0.59	0.48	0.78	0.68	0.60	0.78	0.65	0.56
Lily	0.59	0.56	0.53	0.67	0.02	0.01	0.59	0.32	0.22
LogMap	0.82	0.69	0.59	0.81	0.70	0.62	0.80	0.67	0.57
LogMapLt	0.73	0.59	0.50	0.73	0.67	0.62	0.72	0.67	0.63
ONTMAT1	0.82	0.55	0.41	0.82	0.64	0.52	0.82	0.64	0.53
SANOM	0.79	0.74	0.69	0.66	0.74	0.83	0.65	0.72	0.81
Wiktionary	0.69	0.61	0.54	0.81	0.68	0.58	0.74	0.69	0.64

four systems (AML, Lily, LogMap, Wiktionary) have a wide variation of confidence values.

When comparing the performance of the matchers on the uncertain reference alignments versus that on the sharp version (with the corresponding *ra1*), we see that in the discrete case all matchers except Lily performed the same or better in terms of F-measure (Lily's F-measure dropped almost to 0). Changes in F-measure of discrete cases ranged from -1 to 17 percent over the sharp reference alignment. This was predominantly driven by increased recall, which is a result of the presence of fewer 'controversial' matches in the uncertain version of the reference alignment.

The performance of the matchers with confidence values always 1.0 is very similar regardless of whether a discrete or continuous evaluation methodology is used, because many of the matches they find are the ones that the experts had high agreement about, while the ones they missed were the more controversial matches. AML produces a fairly wide range of confidence values and has the highest F-measure under both the continuous and discrete evaluation methodologies, indicating that this system's confidence evaluation does a good job of reflecting cohesion among experts on this task. Of the remaining systems, three (DOME, LogMap, SANOM) have relatively small drops in F-measure when moving from discrete to continuous evaluation. Lily's performance drops drastically under the discrete and continuous evaluation methodologies. This is because the matcher assigns low confidence values to some matches in which the labels are equivalent strings, which many crowdsourcers agreed with unless there was a compelling technical reason not to. This hurts recall significantly.

Overall, in comparison with last year, the F-measures of most returning matching systems essentially held constant when evaluated against the uncertain reference alignments. The exception was Lily, whose performance in the discrete case decreased dramatically. ONTMAT1 and Wiktionary are two new systems participating in this year. ONTMAT1's performance in both discrete and continuous cases increases 16 percent in terms of F-measure over the sharp reference alignment from 0.55 to 0.64, which it is mainly driven by increased recall. Wiktionary assigns confidence value of 1.0 to the entities with identical strings in two ontologies, while gives confidence value of 0.5 to other possible candidates. From the results, its performance improves significantly from sharp to discrete and continuous cases.

4.5 Disease and Phenotype Track

In the OAEI 2019 phenotype track 8 systems were able to complete at least one of the tasks with a 6 hours timeout. Table 11 shows the evaluation results in the HP-MP and DOID-ORDO matching tasks, respectively.

Since the consensus reference alignments only allow us to assess how systems perform in comparison with one another, the proposed ranking is only a reference. Note that some of the correspondences in the consensus alignment may be erroneous (false positives) because all systems that agreed on it could be wrong (e.g., in erroneous correspondences with equivalent labels, which are not that uncommon in biomedical tasks). In addition, the consensus alignments will not be complete, because there are likely to be correct correspondences that no system is able to find, and there are a number of

Table 11. Results for the HP-MP and DOID-ORDO tasks based on the consensus reference alignment.

System	Time (s)	# Corresp.	# Unique	Scores			Incoherence	
				Prec.	F-m.	Rec.	Unsat.	Degree
HP-MP task								
LogMap	43	2,130	1	0.88	0.85	0.82	0	0.0%
LogMapBio	1,740	2,201	50	0.86	0.85	0.83	0	0.0%
AML	90	2,029	330	0.89	0.84	0.80	0	0.0%
LogMapLt	6	1,370	2	1.00	0.75	0.60	0	0.0%
POMAP++	1,862	1,502	218	0.86	0.68	0.57	0	0.0%
FCAMapKG	14	734	0	1.00	0.49	0.32	0	0.0%
DOME	11	692	0	1.00	0.47	0.30	0	0.0%
Wiktionary	745	61,872	60,634	0.02	0.04	0.55	0	0.0%
DOID-ORDO task								
LogMapBio	2,312	2,547	123	0.91	0.86	0.81	0	0.0%
LogMap	24	2,323	0	0.95	0.85	0.77	0	0.0%
POMAP++	2,497	2,563	192	0.89	0.84	0.79	0	0.0%
LogMapLt	8	1,747	20	0.99	0.75	0.60	0	0.0%
AML	173	4,781	2,342	0.52	0.65	0.87	0	0.0%
FCAMapKG	23	1,274	2	1.00	0.61	0.44	0	0.0%
DOME	17	1,235	5	0.99	0.60	0.43	0	0.0%
Wiktionary	531	909	366	0.57	0.28	0.18	7	0.067%

correspondences found by only one system (and therefore not in the consensus alignments) which may be correct. Nevertheless, the results with respect to the consensus alignments do provide some insights into the performance of the systems.

Overall, LogMap and LogMapBio are the systems that provide the closest set of correspondences to the consensus (not necessarily the best system) in both tasks. LogMap has a small set of unique correspondences as most of its correspondences are also suggested by its variant LogMapBio and vice versa. By contrast, AML and Wiktionary produce the highest number of unique correspondences in HP-MP and DOID-ORDO respectively, and the second-highest inversely. Nonetheless, Wiktionary suggests a very large number of correspondences with respect to the other systems which suggest that it may also include many subsumption and related correspondences and not only equivalence. All systems produce coherent alignments except for Wiktionary in the DOID-ORDO task.

4.6 Large Biomedical Ontologies

In the OAEI 2019 Large Biomedical Ontologies track, 10 systems were able to complete at least one of the tasks within a 6 hours timeout. Eight systems were able to complete all six tasks.[16] The evaluation results for the largest matching tasks are shown in Table 12.

The top-ranked systems by F-measure were respectively: AML and LogMap in Task 2; LogMap and LogMapBio in Task 4; and AML and LogMapBio in Task 6.

[16] Check out the supporting scripts to reproduce the evaluation: `https://github.com/ernestojimenezruiz/oaei-evaluation`

Table 12. Results for the whole ontologies matching tasks in the OAEI largebio track.

System	Time (s)	# Corresp.	# Unique	Scores			Incoherence	
				Prec.	F-m.	Rec.	Unsat.	Degree
Whole FMA and NCI ontologies (Task 2)								
AML	75	3,110	276	0.81	0.84	0.88	4	0.012%
LogMap	82	2,701	0	0.86	0.83	0.81	3	0.009%
LogMapBio	2,072	3,104	139	0.78	0.81	0.85	3	0.009%
LogMapLt	9	3,458	75	0.68	0.74	0.82	8,925	27.3%
Wiktionary	4,699	1,873	56	0.93	0.73	0.61	3,476	10.6%
DOME	21	2,413	7	0.80	0.73	0.67	1,033	3.2%
FCAMapKG	0	3,765	316	0.62	0.71	0.82	10,708	32.8%
AGM	3,325	7,648	6,819	0.08	0.12	0.22	28,537	87.4%
Whole FMA ontology with SNOMED large fragment (Task 4)								
LogMap	394	6,393	0	0.84	0.73	0.65	0	0.0%
LogMapBio	2,853	6,926	280	0.79	0.72	0.67	0	0.0%
AML	152	8,163	2,525	0.69	0.70	0.71	0	0.0%
FCAMapKG	0	1,863	77	0.88	0.36	0.22	1,527	2.0%
LogMapLt	15	1,820	47	0.85	0.33	0.21	1,386	1.8%
DOME	38	1,589	1	0.94	0.33	0.20	1,348	1.8%
Wiktionary	12,633	1,486	143	0.82	0.28	0.17	790	1.0%
AGM	4,227	11,896	10,644	0.07	0.09	0.13	70,923	92.7%
Whole NCI ontology with SNOMED large fragment (Task 6)								
AML	331	14,200	2,656	0.86	0.77	0.69	$\geq$578	$\geq$0.5%
LogMapBio	4,586	13,732	940	0.81	0.71	0.63	$\geq$1	$\geq$0.001%
LogMap	590	12,276	0	0.87	0.71	0.60	$\geq$1	$\geq$0.001%
LogMapLt	16	12,864	658	0.80	0.66	0.57	$\geq$91,207	$\geq$84.7%
FCAMapKG	0	12,813	1,115	0.79	0.65	0.56	$\geq$84,579	$\geq$78.5%
DOME	38	9,806	26	0.91	0.64	0.49	$\geq$66,317	$\geq$61.6%
Wiktionary	9,208	9,585	518	0.90	0.62	0.47	$\geq$65,968	$\geq$61.2%
AGM	5,016	21,600	16,253	0.23	0.25	0.28	-	-

Interestingly, the use of background knowledge led to an improvement in recall from LogMapBio over LogMap in all tasks, but this came at the cost of precision, resulting in the two variants of the system having very similar F-measures.

The effectiveness of all systems decreased from small fragments to whole ontologies tasks.[17] One reason for this is that with larger ontologies there are more plausible correspondence candidates, and thus it is harder to attain both a high precision and a high recall. In fact, this same pattern is observed moving from the FMA-NCI to the FMA-SNOMED to the SNOMED-NCI problem, as the size of the task also increases. Another reason is that the very scale of the problem constrains the matching strategies that systems can employ: AML for example, forgoes its matching algorithms that are computationally more complex when handling very large ontologies, due to efficiency concerns.

[17] http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/2019/results/

The size of the whole ontologies tasks proved a problem for a some of the systems, which were unable to complete them within the allotted time: POMAP++ and SANOM.

With respect to alignment coherence, as in previous OAEI editions, only two distinct systems have shown alignment repair facilities: AML, LogMap and its LogMapBio variant. Note that only LogMap and LogMapBio are able to reduce to a minimum the number of unsatisfiable classes across all tasks, missing 3 unsatisfiable classes in the worst case (whole FMA-NCI task). For the AGM correspondences the ELK reasoner could not complete the classification over the integrated ontology within the allocated time.

As the results tables show, even the most precise alignment sets may lead to a huge number of unsatisfiable classes. This proves the importance of using techniques to assess the coherence of the generated alignments if they are to be used in tasks involving reasoning. We encourage ontology matching system developers to develop their own repair techniques or to use state-of-the-art techniques such as Alcomo [39], the repair module of LogMap (LogMap-Repair) [31] or the repair module of AML [43], which have worked well in practice [33, 24].

4.7 Multifarm

This year, 5 systems registered to participate in the MultiFarm track: AML, EVOCROS, Lily, LogMap and Wiktionary. This number slightly decreases with respect to the last campaign (6 in 2018, 8 in 2017, 7 in 2016, 5 in 2015, 3 in 2014, 7 in 2013, and 7 in 2012). The reader can refer to the OAEI papers for a detailed description of the strategies adopted by each system. In fact, most systems still adopt a translation step before the matching itself. However, a few systems had issues when evaluated: i) EVOCROS encountered problems to complete a single matching task; and ii) Lily has generated mostly empty alignments.

The Multifarm evaluation results based on the blind dataset are presented in Table 13. They have been computed using the Alignment API 4.9 and can slightly differ from those computed with the SEALS client. We haven't applied any threshold on the results. We do not report the results of non-specific systems here, as we could observe in the last campaigns that they can have intermediate results in the "same ontologies" task (ii) and poor performance in the "different ontologies" task (i).

AML outperforms all other systems in terms of F-measure for task i) (same behaviour than last year). In terms of precision, the systems have relatively similar results. With respect to the task ii) LogMap has the best performance. AML and LogMap have participated last year. Comparing the results from last year, in terms F-measure (cases of type i), AML maintains its overall performance (.45 in 2019, .46 in 2018, .46 in 2017, .45 in 2016 and .47 in 2015). The same could be observed for LogMap (.37 in 2018, .36 in 2017, and .37 in 2016).

In terms of performance, the F-measure for blind tests remains relatively stable across campaigns. AML and LogMap keep their positions and have similar F-measure with respect to the previous campaigns. As observed in previous campaigns, systems privilege precision over recall, and the results are expectedly below the ones obtained for the original Conference dataset. Cross-lingual approaches remain mainly based on translation strategies and the combination of other resources (like cross-lingual links

Table 13. MultiFarm aggregated results per matcher, for each type of matching task – different ontologies (i) and same ontologies (ii). Time is measured in minutes (for completing the 55×24 matching tasks); #pairs indicates the number of pairs of languages for which the tool is able to generate (non-empty) alignments; size indicates the average of the number of generated correspondences for the tests where an (non-empty) alignment has been generated. Two kinds of results are reported: those not distinguishing empty and erroneous (or not generated) alignments and those—indicated between parenthesis—considering only non-empty generated alignments for a pair of languages.

System	Time	#pairs	Type (i) – 22 tests per pair				Type (ii) – 2 tests per pair			
			Size	Prec.	F-m.	Rec.	Size	Prec.	F-m.	Rec.
AML	236	55	8.18	.72 (.72)	.45 (.45)	.34 (.34)	33.40	.93 (.95)	.27 (.28)	.17 (.16)
LogMap	49	55	6.99	.72 (.72)	.37 (.37)	.25 (.25)	46.80	.95 (.96)	.41 (.42)	.28 (.28)
Wiktionary	785	23	4.91	.76 (.79)	.31 (.33)	.21 (.22)	9.24	.94 (.96)	.12 (.12)	.07 (.06)

in Wikipedia, BabelNet, etc.) while strategies such as machine learning, or indirect alignment composition remain under-exploited.

4.8 Link Discovery

This year the Link Discovery track counted one participant in the Linking test case (AML) and three participants in the Spatial test case: AML, Silk and RADON. Those were the exact same systems (and versions) that participated on OAEI 2018.

In the Linking test case, AML perfectly captures all the correct links while not producing wrong ones, thus obtaining perfect precision and a recall (1.0) in both the Sandbox and Mainbox datasets. It required 9.7s and 360s, respectively, to complete the two tasks. The results can also be found in HOBBIT platform (`https://tinyurl.com/yywwlsmt` - Login as Guest).

We divided the Spatial test cases into four suites. In the first two suites (SLL and LLL), the systems were asked to match LineStrings to LineStrings considering a given relation for 200 and 2K instances for the TomTom and Spaten datasets. In the last two tasks (SLP, LLP), the systems were asked to match LineStrings to Polygons (or Polygons to LineStrings depending on the relation) again for both datasets. Since the precision, recall and f-measure results from all systems were equal to 1.0, we are only presenting results regarding the time performance. The time performance of the matching systems in the SLL, LLL, SLP and LLP suites are shown in Figures 2-3. The results can also be found in HOBBIT platform (`https://tinyurl.com/y4vk6htq` - Login as Guest).

In the SLL suite, RADON has the best performance in most cases except for the *Touches* and *Intersects* relations, followed by AML. Silk seems to need the most time, particularly for *Touches* and *Intersects* relations in the TomTom dataset and *Overlaps* in both datasets.

In the LLL suite we have a more clear view of the capabilities of the systems with the increase in the number of instances. In this case, RADON and Silk have similar behavior as in the the small dataset, but it is more clear that the systems need much

more time to match instances from the TomTom dataset. RADON has still the best performance in most cases. AML has the next best performance and is able to handle some cases better than other systems (e.g. *Touches* and *Intersects*), however, it also hits the platform time limit in the case of *Disjoint*.

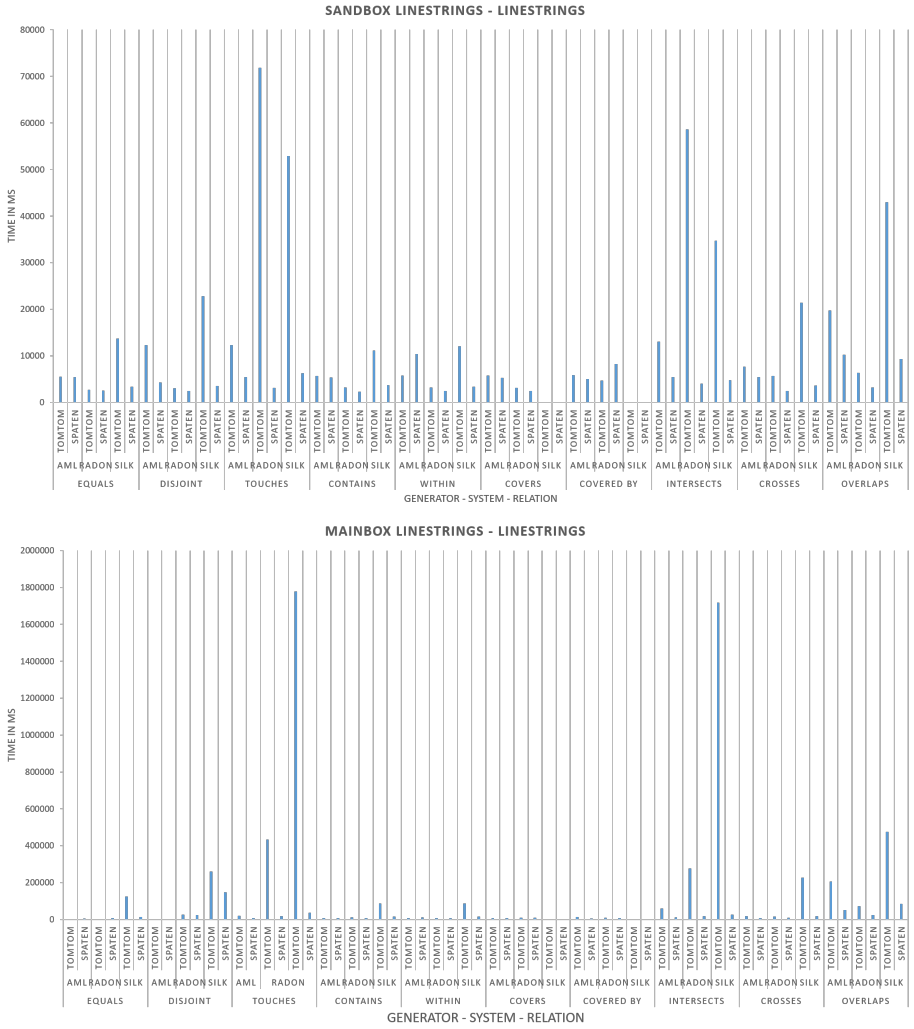


Fig. 2. Time performance for TomTom & Spaten SLL (top) and LLL (bottom) suites for AML (A), Silk (S) and RADON (R).

In the SLP suite, in contrast to the first two suites, RADON has the best performance for all relations. AML and Silk have minor time differences and, depending on the case,

one is slightly better than the other. All the systems need more time for the TomTom dataset but due to the small size of the instances the time difference is minor.

In the LLP suite, RADON again has the best performance in all cases. AML hits the platform time limit in *Disjoint* relations on both datasets and is better than Silk in most cases except *Contains* and *Within* on the TomTom dataset where it needs an excessive amount of time.

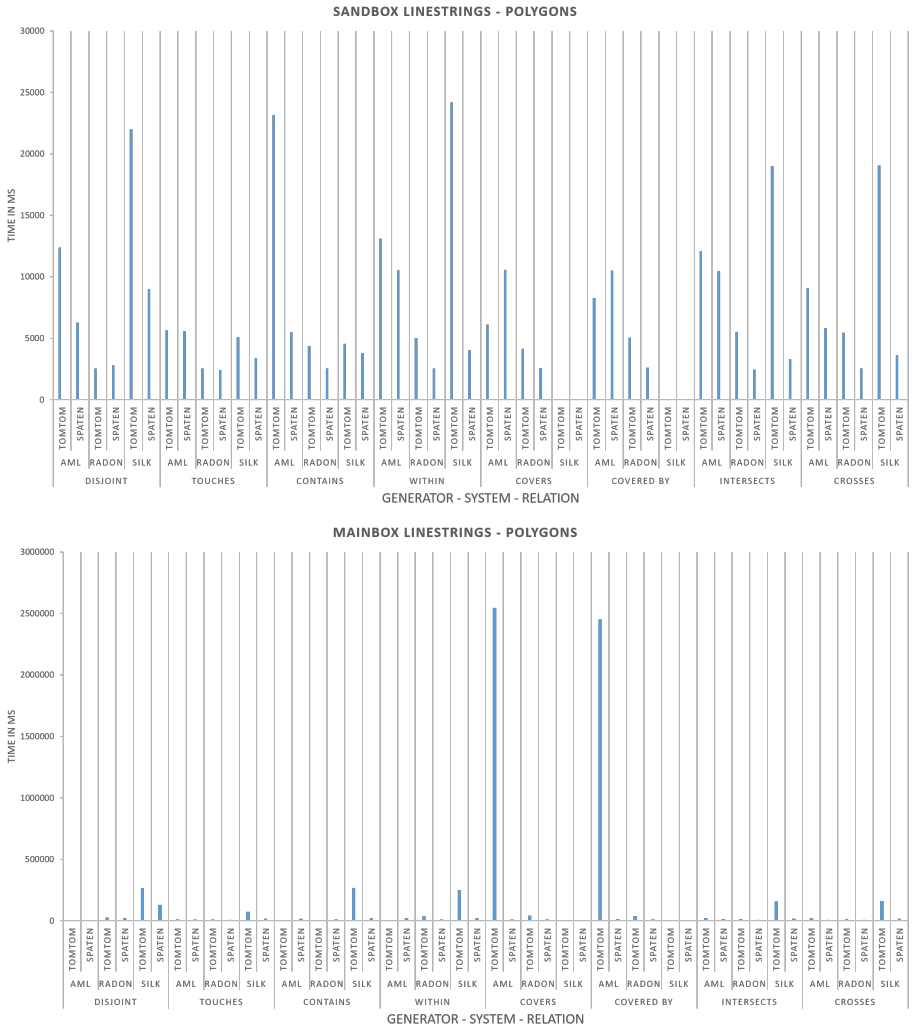


Fig. 3. Time performance for TomTom & Spaten SLP (top) and LLP (bottom) suites for AML (A), Silk (S) and RADON (R).

Taking into account the executed test cases we can identify the capabilities of the tested systems as well as suggest some improvements. All the systems participated in most of the test cases, with the exception of Silk which did not participate in the *Covers* and *Covered By* test cases.

RADON was the only system that successfully addressed all the tasks, and had the best performance for the SLP and LLP suites, but it can be improved for the *Touches* and *Intersects* relations for the SLL and LLL suites. AML performs extremely well in most cases, but can be improved in the cases of *Covers/Covered By* and *Contains/Within* when it comes to LineStrings/Polygons Tasks and especially in *Disjoint* relations where it hits the platform time limit. Silk can be improved for the *Touches*, *Intersects* and *Overlaps* relations and for the SLL and LLL tasks and for the *Disjoint* relation in SLP and LLP Tasks.

In general, all systems needed more time to match the TomTom dataset than the Spaten one, due to the smaller number of points per instance in the latter. Comparing the LineString/LineString to the LineString/Polygon Tasks we can say that all the systems needed less time for the first for the *Contains*, *Within*, *Covers* and *Covered by* relations, more time for the *Touches*, *Instersects* and *Crosses* relations, and approximately the same time for the *Disjoint* relation.

4.9 SPIMBENCH

This year, the SPIMBENCH track counted four participants: AML, Lily, LogMap and FTRLIM. FTRLIM participated for the first time this year while AML, Lily, and LogMap also participated the previous years. The evaluation results of the track are shown in Table 14. The results can also be found in HOBBIT platform (`https://tinyurl.com/yxhsw48c` - Login as Guest).

Table 14. SPIMBENCH track results.

System	Precision	Recall	F-measure	Time (ms)
Sandbox (100 instances)				
AML	0.8348	0.8963	0.8645	6223
Lily	0.8494	1.0	0.9185	2032
LogMap	0.9382	0.7625	0.8413	6919
FTRLIM	0.8542	1.0	0.9214	1474
Mainbox (5000 instances)				
AML	0.8385	0.8835	0.8604	39515
Lily	0.8546	1.0	0.9216	3667
LogMap	0.8925	0.7094	0.7905	26920
FTRLIM	0.8558	1.0	0.9214	2155

Lily and FTRLIM had the best performance overall both in terms of F-measure and run time. Notably, their run time scaled very well with the increase in the number of instances. Lily, FTRLIM, and AML had a higher recall than precision, while Lily and FTRLIM had a full recall. By contrast, LogMap had the highest precision but lowest

recall of all the systems. AML and LogMap had a similar run time for the Sandbox task, but the latter scaled better with the increase in the number of instances.

4.10 Knowledge Graph

We evaluated all SEALS participants in the OAEI (even those not registered for the track) on a very small matching task[18]. This revealed that not all systems were able to handle the task, and in the end, only the following systems were evaluated: AGM, AML, DOME, FCAMap-KG, LogMap, LogMapBio, LogMapKG, LogMapLt, POMap++, Wiktionary. Out of those only LogMapBio, LogMapLt and POMap++ were not registered for this track. In comparison to last year, more matchers participate and return meaningful correspondences. Moreover there are systems which especially focus on the knowledge graph track e.g. FCAMap-KG and LogMapKG.

Table 15 shows the aggregated results for all systems, including the number of tasks in which they were able to generate a non-empty alignment (#tasks) and the average number of generated correspondences in those tasks (size). In addition to the global average precision, F-measure, and recall results, in which tasks where systems produced empty alignments were counted, we also computed F-measure and recall ignoring empty alignments which are shown between parentheses in the table, where applicable.

Nearly all systems were able to generate class correspondences. In terms of F-measure, AML is the best one (when considering only completed test cases). Many matchers were also able to beat the baseline. The highest recall is about 0.77 which shows that some class correspondences are not easy to find.

In comparison to last year, more matchers are able to produce property correspondences. Only the systems of the LogMap family and POMAP++ do not return any alignments. While Wiktionary and FCAMap-KG achieve an F-Measure of 0.98, other systems need more improvement here because they are not capable of beating the baseline (mostly due to low recall).

With respect to instance correspondences, AML and DOME are the best systems, but they outperform the baselines only by a small margin. On average, the systems returned between 3,000 and 8,000 instance alignments. Only LogMapKG returned nearly 30,000 mappings. This is interesting because it should be focused on generating only 1:1 alignments, but deviates here.

We also analyzed the arity of the resulting alignments because in the knowledge graph track it is probably better to focus on a 1:1 mapping. Such a strict mapping is returned by the following systems: AGM, baselineLabel, DOME and POMAP++. LogMap and LogMapBio return a few correspondences with same source or target in only two test cases. BaselineAltLabel, FCAMap-KG and Wiktionary returned some n:m mappings in all test cases. AML and LogMapLt returned more of those and LogMapKG has the highest amount of n:m mappings.

When analyzing the confidence values of the alignments, it turns out that most matchers set it to 1 (AGM,baselineAltLabel, baselineLabel, FCAMap-KG, LogMapLt,

[18] `http://oaei.ontologymatching.org/2019/results/knowledgegraph/small_test.zip`

Table 15. Knowledge Graph track results, divided into class, property, instance, and overall correspondences.

System	Time (s)	# tasks	Size	Prec.	F-m.	Rec.
Class performance						
AGM	10:47:38	5	14.6	0.23	0.09	0.06)
AML	0:45:46	4	27.5	0.78 (0.98)	0.69 (0.86)	0.61 (0.77)
baselineAltLabel	0:11:48	5	16.4	1.0	0.74	0.59
baselineLabel	0:12:30	5	16.4	1.0	0.74	0.59
DOME	1:05:26	4	22.5	0.74 (0.92)	0.62 (0.77)	0.53 (0.66)
FCAMap-KG	1:14:49	5	18.6	1.0	0.82	0.70
LogMap	0:15:43	5	26.0	0.95	0.84	0.76)
LogMapBio	2:31:01	5	26.0	0.95	0.84	0.76)
LogMapKG	2:26:14	5	26.0	0.95	0.84	0.76)
LogMapLt	0:07:28	4	23.0	0.80 (1.0)	0.56 (0.70)	0.43 (0.54)
POMAP++	0:14:39	5	2.0	0.0	0.0	0.0
Wiktionary	0:20:14	5	21.4	1.0	0.8	0.67
Property performance						
AGM	10:47:38	5	49.4	0.66	0.32	0.21)
AML	0:45:46	4	58.2	0.72 (0.91)	0.59 (0.73)	0.49 (0.62)
baselineAltLabel	0:11:48	5	47.8	0.99	0.79	0.66
baselineLabel	0:12:30	5	47.8	0.99	0.79	0.66
DOME	1:05:26	4	75.5	0.79 (0.99)	0.77 (0.96)	0.75 (0.93)
FCAMap-KG	1:14:49	5	69.0	1.0	0.98	0.96
LogMap	0:15:43	5	0.0	0.0	0.0	0.0)
LogMapBio	2:31:01	5	0.0	0.0	0.0	0.0)
LogMapKG	2:26:14	5	0.0	0.0	0.0	0.0)
LogMapLt	0:07:28	4	0.0	0.0	0.0	0.0)
POMAP++	0:14:39	5	0.0	0.0	0.0	0.0)
Wiktionary	0:20:14	5	75.8	0.97	0.98	0.98
Instance performance						
AGM	10:47:38	5	5169.0	0.48	0.25	0.17)
AML	0:45:46	4	7529.8	0.72 (0.90)	0.71 (0.88)	0.69 (0.86)
baselineAltLabel	0:11:48	5	4674.2	0.89	0.84	0.80
baselineLabel	0:12:30	5	3641.2	0.95	0.81	0.71
DOME	1:05:26	4	4895.2	0.74 (0.92)	0.70 (0.88)	0.67 (0.84)
FCAMap-KG	1:14:49	5	4530.6	0.90	0.84	0.79
LogMap	0:15:43	5	0.0	0.0	0.0	0.0)
LogMapBio	2:31:01	5	0.0	0.0	0.0	0.0)
LogMapKG	2:26:14	5	29190.4	0.40	0.54	0.86)
LogMapLt	0:07:28	4	6653.8	0.73 (0.91)	0.67 (0.84)	0.62 (0.78)
POMAP++	0:14:39	5	0.0	0.0	0.0	0.0
Wiktionary	0:20:14	5	3483.6	0.91	0.79	0.70
Overall performance						
AGM	10:47:38	5	5233.2	0.48	0.25	0.17)
AML	0:45:46	4	7615.5	0.72 (0.90)	0.70 (0.88)	0.69 (0.86)
baselineAltLabel	0:11:48	5	4739.0	0.89	0.84	0.80
baselineLabel	0:12:30	5	3706.0	0.95	0.81	0.71
DOME	1:05:26	4	4994.8	0.74 (0.92)	0.70 (0.88)	0.67 (0.84)
FCAMap-KG	1:14:49	5	4792.6	0.91	0.85	0.79
LogMap	0:15:43	5	26.0	0.95	0.01	0.0)
LogMapBio	2:31:01	5	26.0	0.95	0.01	0.0)
LogMapKG	2:26:14	5	29216.4	0.40	0.54	0.84)
LogMapLt	0:07:28	4	6676.8	0.73 (0.91)	0.66 (0.83)	0.61 (0.76)
POMAP++	0:14:39	5	19.4	0.0	0.0	0.0
Wiktionary	0:20:14	5	3581.8	0.91	0.8	0.71

Wiktionary). AML and LogMapKG set it higher than 0.6 whereas only DOME uses the full range between zero and one. LogMap and LogMapBio uses a range of 0.3 and 0.8. The confidences were analyzed with the MELT dashboard[19] [28].

Regarding runtime, AGM (10:47:38) was the slowest system, followed by LogMapKG and LogMapBio which were much faster. Besides AGM all five test cases could be completed in under 3 hours.

4.11 Interactive matching

This year, three systems participated in the Interactive matching track. They are ALIN, AML, and LogMap. Their results are shown in Table 16 and Figure 4 for both Anatomy and Conference datasets.

The table includes the following information (column names within parentheses):

- The performance of the system: Precision (Prec.), Recall (Rec.) and F-measure (F-m.) with respect to the fixed reference alignment, as well as Recall+ (Rec.+) for the Anatomy task. To facilitate the assessment of the impact of user interactions, we also provide the performance results from the original tracks, without interaction (line with Error NI).
- To ascertain the impact of the oracle errors, we provide the performance of the system with respect to the oracle (i.e., the reference alignment as modified by the errors introduced by the oracle: Precision oracle (Prec. oracle), Recall oracle (Rec. oracle) and F-measure oracle (F-m. oracle). For a perfect oracle these values match the actual performance of the system.
- Total requests (Tot Reqs.) represents the number of distinct user interactions with the tool, where each interaction can contain one to three conflicting correspondences, that could be analysed simultaneously by a user.
- Distinct correspondences (Dist. Mapps) counts the total number of correspondences for which the oracle gave feedback to the user (regardless of whether they were submitted simultaneously, or separately).
- Finally, the performance of the oracle itself with respect to the errors it introduced can be gauged through the positive precision (Pos. Prec.) and negative precision (Neg. Prec.), which measure respectively the fraction of positive and negative answers given by the oracle that are correct. For a perfect oracle these values are equal to 1 (or 0, if no questions were asked).

The figure shows the time intervals between the questions to the user/oracle for the different systems and error rates. Different runs are depicted with different colors.

The matching systems that participated in this track employ different user-interaction strategies. While LogMap, and AML make use of user interactions exclusively in the post-matching steps to filter their candidate correspondences, ALIN can also add new candidate correspondences to its initial set. LogMap and AML both request feedback on only selected correspondences candidates (based on their similarity

[19] `http://oaei.ontologymatching.org/2019/results/knowledgegraph/knowledge_graph_dashboard.html`

Table 16. Interactive matching results for the Anatomy and Conference datasets.

Tool	Error	Prec.	Rec.	F-m.	Rec.+	Prec. oracle	Rec. oracle	F-m. oracle	Tot. Reqs.	Dist. Mapps	Pos. Prec.	Neg. Prec.
Anatomy Dataset												
ALIN	NI	0.974	0.698	0.813	0.365	–	–	–	–	–	–	–
	0.0	0.979	0.85	0.91	0.63	0.979	0.85	0.91	365	638	1.0	1.0
	0.1	0.953	0.832	0.889	0.599	0.979	0.848	0.909	339	564	0.854	0.933
	0.2	0.929	0.817	0.869	0.569	0.979	0.848	0.909	332	549	0.728	0.852
	0.3	0.908	0.799	0.85	0.54	0.979	0.847	0.908	326	536	0.616	0.765
AML	NI	0.95	0.936	0.943	0.832	–	–	–	–	–	–	–
	0.0	0.968	0.948	0.958	0.862	0.968	0.948	0.958	236	235	1.0	1.0
	0.1	0.954	0.944	0.949	0.853	0.969	0.947	0.958	237	235	0.696	0.973
	0.2	0.944	0.94	0.942	0.846	0.969	0.948	0.959	252	248	0.565	0.933
	0.3	0.935	0.933	0.933	0.827	0.969	0.946	0.957	238	234	0.415	0.878
LogMap	NI	0.918	0.846	0.88	0.593	–	–	–	–	–	–	–
	0.0	0.982	0.846	0.909	0.595	0.982	0.846	0.909	388	1164	1.0	1.0
	0.1	0.962	0.831	0.892	0.566	0.964	0.803	0.876	388	1164	0.752	0.965
	0.2	0.945	0.822	0.879	0.549	0.945	0.763	0.844	388	1164	0.57	0.926
	0.3	0.933	0.815	0.87	0.535	0.921	0.724	0.811	388	1164	0.432	0.872
Conference Dataset												
ALIN	NI	0.871	0.443	0.587	–	–	–	–	–	–	–	–
	0.0	0.914	0.695	0.79	–	0.914	0.695	0.79	228	373	1.0	1.0
	0.1	0.809	0.658	0.725	–	0.919	0.704	0.798	226	367	0.707	0.971
	0.2	0.715	0.631	0.67	–	0.926	0.717	0.808	221	357	0.5	0.942
	0.3	0.636	0.605	0.62	–	0.931	0.73	0.819	219	353	0.366	0.908
AML	NI	0.841	0.659	0.739		–	–	–	–	–	–	–
	0.0	0.91	0.698	0.79	–	0.91	0.698	0.79	221	220	1.0	1.0
	0.1	0.846	0.687	0.758	–	0.916	0.716	0.804	242	236	0.726	0.971
	0.2	0.783	0.67	0.721	–	0.924	0.729	0.815	263	251	0.571	0.933
	0.3	0.721	0.646	0.681	–	0.927	0.741	0.824	273	257	0.446	0.877
LogMap	NI	0.818	0.59	0.686	–	–	–	–	–	–	–	–
	0.0	0.886	0.61	0.723	–	0.886	0.61	0.723	82	246	1.0	1.0
	0.1	0.845	0.595	0.698	–	0.857	0.576	0.689	82	246	0.694	0.973
	0.2	0.818	0.586	0.683	–	0.827	0.546	0.657	82	246	0.507	0.941
	0.3	0.799	0.588	0.677	–	0.81	0.519	0.633	82	246	0.376	0.914

NI stands for non-interactive, and refers to the results obtained by the matching system in the original track.

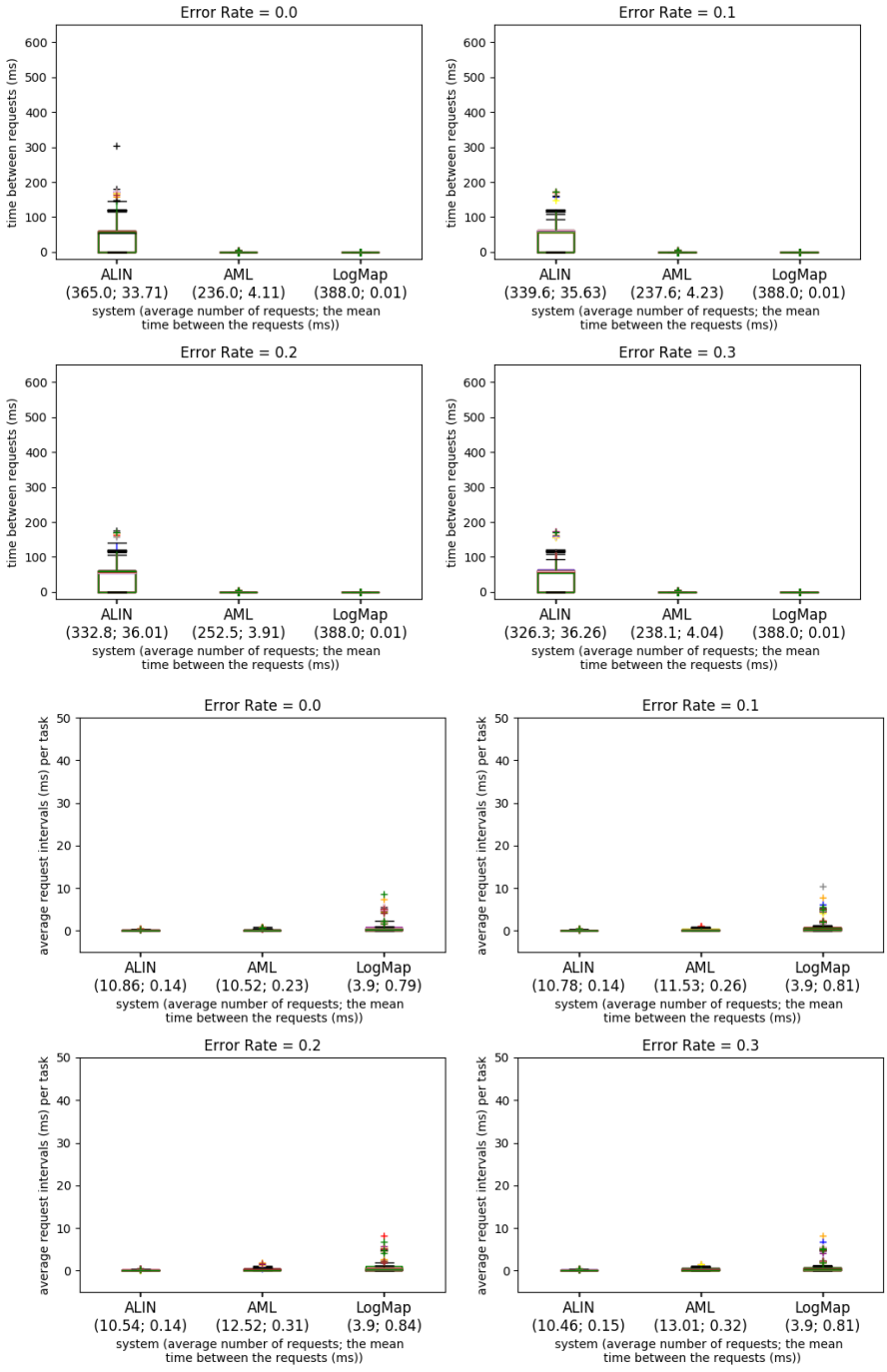


Fig. 4. Time intervals between requests to the user/oracle for the Anatomy (top 4 plots) and Conference (bottom 4 plots) datasets. Whiskers: Q1-1,5IQR, Q3+1,5IQR, IQR=Q3-Q1. The labels under the system names show the average number of requests and the mean time between the requests for the ten runs.

patterns or their involvement in unsatisfiabilities) and AML presents one correspondence at a time to the user. ALIN and LogMap can both ask the oracle to analyze several conflicting correspondences simultaneously.

The performance of the systems usually improves when interacting with a perfect oracle in comparison with no interaction. ALIN is the system that improves the most, because its high number of oracle requests and its non-interactive performance was the lowest of the interactive systems, and thus the easiest to improve.

Although system performance deteriorates when the error rate increases, there are still benefits from the user interaction—some of the systems' measures stay above their non-interactive values even for the larger error rates. Naturally, the more a system relies on the oracle, the more its performance tends to be affected by the oracle's errors.

The impact of the oracle's errors is linear for ALIN, and AML in most tasks, as the F-measure according to the oracle remains approximately constant across all error rates. It is supra-linear for LogMap in all datasets.

Another aspect that was assessed, was the response time of systems, i.e., the time between requests. Two models for system *response times* are frequently used in the literature [11]: Shneiderman and Seow take different approaches to categorize the response times taking a task-centered view and a user-centered view respectively. According to task complexity, Shneiderman defines response time in four categories: typing, mouse movement (50-150 ms), simple frequent tasks (1 s), common tasks (2-4 s) and complex tasks (8-12 s). While Seow's definition of response time is based on the user expectations towards the execution of a task: instantaneous (100-200 ms), immediate (0.5-1 s), continuous (2-5 s), captive (7-10 s). Ontology alignment is a cognitively demanding task and can fall into the third or fourth categories in both models. In this regard the response times (request intervals as we call them above) observed in all datasets fall into the tolerable and acceptable response times, and even into the first categories, in both models. The request intervals for AML, LogMap and XMAP stay at a few milliseconds for most datasets. ALIN's request intervals are higher, but still in the tenth of second range. It could be the case, however, that a user would not be able to take advantage of these low response times because the task complexity may result in higher user response time (i.e., the time the user needs to respond to the system after the system is ready).

4.12 Complex Matching

Three systems were able to generate complex correspondences: AMLC, AROA [53], and CANARD. The results for the other systems are reported in terms of simple alignments. The results of the systems on the five test cases are summarized in Table 17.

With respect to the Hydrography test case, only AMLC can generate two correct complex correspondences which are stating that a class in the source ontology is equivalent to the union of two classes in the target ontology. Most of the systems achieved fair results in terms of precision, but the low recall reflects that the current ontology alignment systems still need to be improved to find more complex relations.

In terms of GeoLink test cases, the real-world instance data from GeoLink Project is also populated into the ontology in order to enable the systems that depend on instance-based matching algorithms to evaluate their performance. There are three alignment

Table 17. Results of the Complex Track in OAEI 2019.

Matcher	Conference			Populated Conference		Hydrography			GeoLink			Taxon	
	Prec.	F-meas.	Rec.	Prec.	Coverage	relaxed_Prec.	relaxed_F-meas.	relaxed_Rec.	relaxed_Prec.	relaxed_F-meas.	relaxed_Rec.	Prec.	Coverage
AGM	-	-	-	-	-	-	-	-	-	-	-	0.06 - 0.14	0.03 - 0.04
Alin	-	-	-	**0.68 - 0.98**	0.20 - 0.28	-	-	-	-	-	-	-	-
AML	-	-	-	0.59 - 0.93	0.31 - 0.37	-	-	-	-	-	-	0.53	0.00
AMLC	**0.31**	**0.34**	**0.37**	0.30 - 0.59	**0.46 - 0.50**	0.45	0.10	0.05	0.50	0.32	0.23	-	-
AROA	-	-	-	-	-	-	-	-	0.87	**0.60**	**0.46**	-	-
CANARD	-	-	-	0.21 - 0.88	0.40 - **0.51**	-	-	-	0.89	0.54	0.39	0.08 - 0.91	**0.14 - 0.36**
DOME	-	-	-	0.59 - 0.94	0.40 - 0.51	-	-	-	-	-	-	-	-
FcaMapKG	-	-	-	0.51 - 0.82	0.21 - 0.28	-	-	-	-	-	-	0.63 - 0.96	0.03 - 0.05
Lily	-	-	-	0.45 - 0.73	0.23 - 0.28	-	-	-	-	-	-	-	-
LogMap	-	-	-	**0.56 - 0.96**	0.25 - 0.32	0.67	0.10	0.05	0.85	0.29	0.18	0.63 - 0.79	0.11 - 0.14
LogMapBio	-	-	-	-	-	**0.70**	0.10	0.05	-	-	-	0.54 - 0.72	0.08 - 0.11
LogMapKG	-	-	-	**0.56 - 0.96**	0.25 - 0.32	0.67	0.10	0.05	-	-	-	0.55 - 0.69	**0.14** - 0.17
LogMapLt	-	-	-	0.50 - 0.87	0.23 - 0.32	0.67	0.10	0.05	-	-	-	0.54 - 0.72	0.08 - 0.11
ONTMAT1	-	-	-	**0.67 - 0.98**	0.20 - 0.28	-	-	-	-	-	-	-	-
POMAP++	-	-	-	0.25 - 0.54	0.20 - 0.29	0.65	0.07	0.04	**0.90**	0.26	0.16	1.00	0.00
Wikitionary	-	-	-	0.48 - 0.88	0.26 - 0.34	-	-	-	-	-	-	-	-

systems that generate complex alignments in GeoLink Benchmark, which are AMLC, AROA, and CANARD. AMLC didn't find any correct complex alignment, while AROA and CARARD achieved relatively good performance. One of the reasons may be that these two systems are instance-based systems, which rely on the shared instances between ontologies. In other words, the shared instance data between two ontologies would be helpful to the matching process.

In the Taxon test cases, only the output of LogMap, LogMapLt and CANARD could be used to rewrite source queries.

With respect to the Conference test cases although the performance in terms of precision and recall decreased for AMLC, AMLC managed to find more true positives than the last year. Since AMLC provides confidence, it could be possible to include confidence into the evaluation and this could improve the performance results. AMLC discovered one more kind of complex mappings: the union of classes.

A more detailed discussion of the results of each task can be found in the OAEI page for this track. For a second edition of complex matching in an OAEI campaign, and given the inherent difficulty of the task, the results and participation are promising albeit still modest.

5 Conclusions & Lessons Learned

In 2019, we witnessed a slight decrease in the number of participants in comparison with previous years, but with a healthy mix of new and returning systems. However, like last year, the distribution of participants by tracks was uneven.

The **schema matching tracks** saw abundant participation, but, as has been the trend of the recent years, little substantial progress in terms of quality of the results or run time of top matching systems, judging from the long-standing tracks. On the one hand, this may be a sign of a performance plateau being reached by existing strategies and algorithms, which would suggest that new technology is needed to obtain significant improvements. On the other hand, it is also true that established matching systems tend to focus more on new tracks and datasets than on improving their performance in long-standing tracks, whereas new systems typically struggle to compete with established ones.

The number of matching systems capable of handling very large ontologies has increased slightly over the last years, but is still relatively modest, judging from the *Large Biomedical Ontologies* track. We will aim at facilitating participation in future editions of this track by providing techniques to divide the matching tasks in manageable subtasks (e.g., [30]).

According to the *Conference* track there is still need for an improvement with regard to the ability of matching systems to match properties. To assist system developers in tackling this aspect we provided a more detailed evaluation in terms of the analysis of the false positives per matching system (available on the Conference track web page). However, this could be extended by the inspection of the reasons why the matching system found the given false positives.As already pointed out last year, less encouraging is the low number of systems concerned with the logical coherence of the alignments they produce, an aspect which is critical for several semantic web applications. Perhaps a more direct approach is needed to promote this topic, such as providing a more in-depth analysis of the causes of incoherence in the evaluation or even organizing a future track focusing on logical coherence alone.

The consensus-based evaluation in the *Disease and Phenotype* track offers limited insights into performance, as several matching systems produce a number of unique correspondences which may or may not be correct. In the absence of a true reference alignment, future evaluation should seek to determine whether the unique correspondences contain indicators of correctness, such as semantic similarity, or appear to be noise.

Despite the quite promising results obtained by matching systems for the **Biodiversity and Ecology track**, the most important observation is that none of the systems has been able to detect mappings established by the experts. Detecting such correspondences requires the use of domain-specific core knowledge that captures biodiversity concepts. We expect this domain-specific background to be integrated in future versions of the systems.

The **interactive matching track** also witnessed a small number of participants. Three systems participated this year. This is puzzling considering that this track is based on the *Anatomy* and *Conference* test cases, and those tracks had 13 participants. The process of programmatically querying the Oracle class used to simulate user interactions is simple enough that it should not be a deterrent for participation, but perhaps we should look at facilitating the process further in future OAEI editions by providing implementation examples.

The **complex matching track** opens new perspectives in the field of ontology matching. Tackling complex matching automatically is extremely challenging, likely requiring profound adaptations from matching systems, so the fact that there were three participants that were able to generate complex correspondences in this track should be seen as a positive sign of progress to the state of the art in ontology matching. This year automatic evaluation has been introduced following an instance-based comparison approach.

The **instance matching tracks** and the new **instance and schema matching track** counted few participants, as has been the trend in recent years. Part of the reason for this is that several of these tracks ran on the HOBBIT platform, and the transition

from SEALS to HOBBIT has not been as easy as we might desire. Thus, participation should increase next year as systems become more familiar with the HOBBIT platform and have more time to do the migration. Furthermore, from an infrastructure point of view, the HOBBIT SDK will make the developing and debugging phase easier, and the Maven-based framework will facilitate submission. However, another factor behind the reduced participation in the instance matching tracks lies with their specialization. New schema matching tracks such as *Biodiversity and Ecology* typically demand very little from systems that are already able to tackle long-standing tracks such as *Anatomy*, whereas instance matching tracks such as *Link Discovery* and last year's *Process Model Matching*, are so different from one another that each requires dedicated development time to tackle. Thus, in future OAEI editions we should consider publishing new instance matching (and other more specialized) datasets with more time in advance, to give system developers adequate time to tackle them. Equally critical will be to ensure stability by maintaining instance matching tracks and datasets over multiple OAEI editions, so that participants can build upon the development of previous years.

Automatic instance-matching benchmark generation algorithms have been gaining popularity, as evidenced by the fact that they are used in all three instance matching tracks of this OAEI edition. One aspect that has not been addressed in such algorithms is that, if the transformation is too extreme, the correspondence may be unrealistic and impossible to detect even by humans. As such, we argue that *human-in-the-loop* techniques can be exploited to do a preventive quality-checking of generated correspondences, and refine the set of correspondences included in the final reference alignment.

In the **knowledge graph track**, we could observe that simple baselines are still hard to beat – which was also the case in other tracks when they were still new. We expect more sophisticated and powerful implementations in the next editions.

Like in previous OAEI editions, most participants provided a description of their systems and their experience in the evaluation, in the form of OAEI system papers. These papers, like the present one, have not been peer reviewed. However, they are full contributions to this evaluation exercise, reflecting the effort and insight of matching systems developers, and providing details about those systems and the algorithms they implement.

The Ontology Alignment Evaluation Initiative will strive to remain a reference to the ontology matching community by improving both the test cases and the testing methodology to better reflect actual needs, as well as to promote progress in this field. More information can be found at: `http://oaei.ontologymatching.org`.

Acknowledgements

We warmly thank the participants of this campaign. We know that they have worked hard to have their matching tools executable in time and they provided useful reports on their experience. The best way to learn about the results remains to read the papers that follow.

We are grateful to the Universidad Politécnica de Madrid (UPM), especially to Nandana Mihindukulasooriya and Asunción Gómez Pérez, for moving, setting up and providing the necessary infrastructure to run the SEALS repositories.

We are also grateful to Martin Ringwald and Terry Hayamizu for providing the reference alignment for the anatomy ontologies and thank Elena Beisswanger for her thorough support on improving the quality of the dataset.

We thank Andrea Turbati and the AGROVOC team for their very appreciated help with the preparation of the AGROVOC subset ontology. We are also grateful to Catherine Roussey and Nathalie Hernandez for their help on the Taxon alignment.

We also thank for their support the past members of the Ontology Alignment Evaluation Initiative steering committee: Jérôme Euzenat (INRIA, FR), Yannis Kalfoglou (Ricoh laboratories, UK), Miklos Nagy (The Open University,UK), Natasha Noy (Google Inc., USA), Yuzhong Qu (Southeast University, CN), York Sure (Leibniz Gemeinschaft, DE), Jie Tang (Tsinghua University, CN), Heiner Stuckenschmidt (Mannheim Universität, DE), George Vouros (University of the Aegean, GR).

Cássia Trojahn dos Santos has been partially supported by the CNRS Blanc project RegleX-LD.

Daniel Faria was supported by the EC H2020 grant 676559 ELIXIR-EXCELERATE and the Portuguese FCT Grant 22231 BioData.pt, co-financed by FEDER.

Ernesto Jimenez-Ruiz has been partially supported by the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project no.: 237889) and the AIDA project (Alan Turing Institute).

Catia Pesquita was supported by the FCT through the LASIGE Strategic Project (UID/CEC/00408/2013) and the research grant PTDC/EEI-ESS/4633/2014.

Irini Fundulaki and Tzanina Saveta were supported by the EU's Horizon 2020 research and innovation programme under grant agreement No 688227 (Hobbit).

Jana Vataščinová and Ondřej Zamazal were supported by the CSF grant no. 18-23964S.

Patrick Lambrix and Huanyu Li have been supported by the Swedish e-Science Research Centre (SeRC), the Swedish Research Council (Vetenskapsrådet, dnr 2018-04147) and the Swedish National Graduate School in Computer Science (CUGS).

The Biodiversity and Ecology track has been partially funded by the German Research Foundation in the context of the GFBio Project (grant No. SE 553/7-1) and the CRC 1076 AquaDiva, the Leitprojekt der Fraunhofer Gesellschaft in the context of the MED2ICIN project (grant No. 600628) and the German Network for Bioinformatics Infrastructure - de.NBI (grant No. 031A539B).

References

1. Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Ian Harrow, Valentina Ivanova, Ernesto Jiménez-Ruiz, Kristian Kolthoff, Elena Kuss, Patrick Lambrix, Henrik Leopold, Huanyu Li, Christian Meilicke, Majid Mohammadi, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Heiner Stuckenschmidt, Élodie Thiéblin, Konstantin Todorov, Cássia Trojahn, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative

2017. In *Proceedings of the 12th International Workshop on Ontology Matching, Vienna, Austria*, pages 61–113, 2017.

2. Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jerome Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Ian Harrow, Valentina Ivanova, Ernesto Jiménez-Ruiz, Elena Kuss, Patrick Lambrix, Henrik Leopold, Huanyu Li, Christian Meilicke, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Andrea Splendiani, Heiner Stuckenschmidt, Konstantin Todorov, Cássia Trojahn, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative 2016. In *Proceedings of the 11th International Ontology matching workshop, Kobe (JP)*, pages 73–129, 2016.
3. José Luis Aguirre, Bernardo Cuenca Grau, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Robert Willem van Hague, Laura Hollink, Ernesto Jiménez-Ruiz, Christian Meilicke, Andriy Nikolov, Dominique Ritze, François Scharffe, Pavel Shvaiko, Ondrej Sváb-Zamazal, Cássia Trojahn, and Benjamin Zapilko. Results of the ontology alignment evaluation initiative 2012. In *Proceedings of the 7th International Ontology matching workshop, Boston (MA, US)*, pages 73–115, 2012.
4. Alsayed Algergawy, Michelle Cheatham, Daniel Faria, Alfio Ferrara, Irini Fundulaki, Ian Harrow, Sven Hertling, Ernesto Jiménez-Ruiz, Naouel Karam, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Stefano Montanelli, Heiko Paulheim, Catia Pesquita, Tzanina Saveta, Daniela Schmidt, Pavel Shvaiko, Andrea Splendiani, Élodie Thiéblin, Cássia Trojahn, Jana Vatascinová, Ondrej Zamazal, and Lu Zhou. Results of the ontology alignment evaluation initiative 2018. In *Proceedings of the 13th International Workshop on Ontology Matching, Monterey (CA, US)*, pages 76–116, 2018.
5. Benhamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proc. K-Cap Workshop on Integrating Ontologies*, Banff (Canada), 2005.
6. Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.
7. Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Vojtech Svátek. Results of the ontology alignment evaluation initiative 2008. In *Proceedings of the 3rd Ontology matching workshop, Karlsruhe (DE)*, pages 73–120, 2008.
8. Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Patrick Lambrix, Stefano Montanelli, Catia Pesquita, Tzanina Saveta, Pavel Shvaiko, Alessandro Solimando, Cássia Trojahn, and Ondřej Zamazal. Results of the ontology alignment evaluation initiative 2015. In *Proceedings of the 10th International Ontology matching workshop, Bethlehem (PA, US)*, pages 60–115, 2015.
9. Michelle Cheatham, Dalia Varanka, Fatima Arauz, and Lu Zhou. Alignment of surface water ontologies: a comparison of manual and automated approaches. *Journal of Geographical Systems*, pages 1–23, 2019.
10. Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, François Scharffe, Pavel Shvaiko, Cássia Trojahn dos Santos, and Ondrej Zamazal. Results of the ontology alignment evaluation initiative 2013. In Pavel Shvaiko, Jérôme Euzenat, Kavitha Srinivas, Ming Mao, and Ernesto Jiménez-Ruiz, editors, *Proceedings of the 8th International Ontology matching workshop, Sydney (NSW, AU)*, pages 61–100, 2013.
11. Jim Dabrowski and Ethan V. Munson. 40 years of searching for the best computer system response time. *Interacting with Computers*, 23(5):555–564, 2011.

12. Thaleia Dimitra Doudali, Ioannis Konstantinou, and Nectarios Koziris Doudali. Spaten: a Spatio-Temporal and Textual Big Data Generator. In *IEEE Big Data*, pages 3416–3421, 2017.
13. Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, Stefano Montanelli, Heiko Paulheim, Dominique Ritze, Pavel Shvaiko, Alessandro Solimando, Cássia Trojahn dos Santos, Ondrej Zamazal, and Bernardo Cuenca Grau. Results of the ontology alignment evaluation initiative 2014. In *Proceedings of the 9th International Ontology matching workshop, Riva del Garda (IT)*, pages 61–104, 2014.
14. Zlatan Dragisic, Valentina Ivanova, Patrick Lambrix, Daniel Faria, Ernesto Jiménez-Ruiz, and Catia Pesquita. User validation in ontology alignment. In *Proceedings of the 15th International Semantic Web Conference, Kobe (JP)*, pages 200–217, 2016.
15. Zlatan Dragisic, Valentina Ivanova, Huanyu Li, and Patrick Lambrix. Experiences from the anatomy track in the ontology alignment evaluation initiative. *Journal of Biomedical Semantics*, 8:56:1–56:28, 2017.
16. Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In *Integrating Ontologies, Proceedings of the K-CAP Workshop on Integrating Ontologies, Banff, Canada*, 2005.
17. Jérôme Euzenat, Alfio Ferrara, Laura Hollink, Antoine Isaac, Cliff Joslyn, Véronique Malaisé, Christian Meilicke, Andriy Nikolov, Juan Pane, Marta Sabou, François Scharffe, Pavel Shvaiko, Vassilis Spiliopoulos, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Cássia Trojahn dos Santos, George Vouros, and Shenghui Wang. Results of the ontology alignment evaluation initiative 2009. In *Proceedings of the 4th International Ontology matching workshop, Chantilly (VA, US)*, pages 73–126, 2009.
18. Jérôme Euzenat, Alfio Ferrara, Christian Meilicke, Andriy Nikolov, Juan Pane, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, and Cássia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2010. In *Proceedings of the 5th International Ontology matching workshop, Shanghai (CN)*, pages 85–117, 2010.
19. Jérôme Euzenat, Alfio Ferrara, Robert Willem van Hague, Laura Hollink, Christian Meilicke, Andriy Nikolov, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Cássia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2011. In *Proceedings of the 6th International Ontology matching workshop, Bonn (DE)*, pages 85–110, 2011.
20. Jérôme Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. In *Proceedings 2nd International Ontology matching workshop, Busan (KR)*, pages 96–132, 2007.
21. Jérôme Euzenat, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, and Cássia Trojahn dos Santos. Ontology alignment evaluation initiative: six years of experience. *Journal on Data Semantics*, XV:158–192, 2011.
22. Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2006. In *Proceedings of the 1st International Ontology matching workshop, Athens (GA, US)*, pages 73–95, 2006.
23. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, 2nd edition, 2013.
24. Daniel Faria, Ernesto Jiménez-Ruiz, Catia Pesquita, Emanuel Santos, and Francisco M. Couto. Towards Annotating Potential Incoherences in BioPortal Mappings. In *Proceedings of the 13th International Semantic Web Conference*, volume 8797, pages 17–32, 2014.

25. Ian Harrow, Ernesto Jiménez-Ruiz, Andrea Splendiani, Martin Romacker, Peter Woollard, Scott Markel, Yasmin Alam-Faruque, Martin Koch, James Malone, and Arild Waaler. Matching Disease and Phenotype Ontologies in the Ontology Alignment Evaluation Initiative. *Journal of Biomedical Semantics*, 8:55:1–55:13, 2017.
26. Sven Hertling and Heiko Paulheim. Dbkwik: A consolidated knowledge graph from thousands of wikis. In *Proceedings of the International Conference on Big Knowledge*, 2018.
27. Sven Hertling and Heiko Paulheim. Dbkwik: extracting and integrating knowledge from thousands of wikis. *Knowledge and Information Systems*, 2019.
28. Sven Hertling, Jan Portisch, and Heiko Paulheim. Melt - matching evaluation toolkit. In *SEMANTICS*, 2019.
29. Valentina Ivanova, Patrick Lambrix, and Johan Åberg. Requirements for and evaluation of user support for large-scale ontology alignment. In *Proceedings of the European Semantic Web Conference*, pages 3–20, 2015.
30. Ernesto Jiménez-Ruiz, Asan Agibetov, Matthias Samwald, and Valerie Cross. Breaking-down the Ontology Alignment Task with a Lexical Index and Neural Embeddings. *CoRR*, abs/1805.12402, 2018.
31. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In *Proceedings of the 10th International Semantic Web Conference, Bonn (DE)*, pages 273–288, 2011.
32. Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga. Logic-based assessment of the compatibility of UMLS ontology sources. *J. Biomed. Sem.*, 2, 2011.
33. Ernesto Jiménez-Ruiz, Christian Meilicke, Bernardo Cuenca Grau, and Ian Horrocks. Evaluating mapping repair systems with large biomedical ontologies. In *Proceedings of the 26th Description Logics Workshop*, 2013.
34. Ernesto Jiménez-Ruiz, Tzanina Saveta, Ondvrej Zamazal, Sven Hertling, Michael Röder, Irini Fundulaki, Axel-Cyrille Ngonga Ngomo, Mohamed Ahmed Sherif, Amina Annane, Zohra Bellahsene, Sadok Ben Yahia, Gayo Diallo, Daniel Faria, Marouen Kachroudi, Abderrahmane Khiat, Patrick Lambrix, Huanyu Li, Maximilian Mackeprang, Majid Mohammadi, Maciej Rybinski, Booma Sowkarthiga Balasubramani, and Cassia Trojahn. Introducing the HOBBIT platform into the Ontology Alignment Evaluation Campaign. In *Proceedings of the 13th International Workshop on Ontology Matching*, 2018.
35. Naouel Karam, Claudia Müller-Birn, Maren Gleisberg, David Fichtmüller, Robert Tolksdorf, and Anton Güntsch. A terminology service supporting semantic annotation, integration, discovery and analysis of interdisciplinary research data. *Datenbank-Spektrum*, 16(3):195–205, 2016.
36. Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. Concurrent classification of EL ontologies. In *Proceedings of the 10th International Semantic Web Conference, Bonn (DE)*, pages 305–320, 2011.
37. Friederike Klan, Erik Faessler, Alsayed Algergawy, Birgitta König-Ries, and Udo Hahn. Integrated semantic search on structured and unstructured data in the adonis system. In *Proceedings of the 2nd International Workshop on Semantics for Biodiversity*, 2017.
38. Huanyu Li, Zlatan Dragisic, Daniel Faria, Valentina Ivanova, Ernesto Jiménez-Ruiz, Patrick Lambrix, and Catia Pesquita. User validation in ontology alignment: functional assessment and impact. *The Knowledge Engineering Review*, 34:e15, 2019.
39. Christian Meilicke. *Alignment Incoherence in Ontology Matching*. PhD thesis, University Mannheim, 2011.
40. Christian Meilicke, Raúl García Castro, Frederico Freitas, Willem Robert van Hage, Elena Montiel-Ponsoda, Ryan Ribeiro de Azevedo, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Andrei Tamilin, Cássia Trojahn, and Shenghui Wang. MultiFarm: A benchmark for multilingual ontology matching. *Journal of web semantics*, 15(3):62–68, 2012.

41. Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
42. Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards evaluating interactive ontology matching tools. In *Proceedings of the 10th Extended Semantic Web Conference, Montpellier (FR)*, pages 31–45, 2013.
43. Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco M Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PLoS ONE*, 10(12):e0144807, 2015.
44. Tzanina Saveta, Evangelia Daskalaki, Giorgos Flouris, Irini Fundulaki, Melanie Herschel, and Axel-Cyrille Ngonga Ngomo. Pushing the limits of instance matching systems: A semantics-aware benchmark for linked data. In *Proceedings of the 24th International Conference on World Wide Web*, pages 105–106, New York, NY, USA, 2015. ACM.
45. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In *Proceedings of the International Semantic Web Conference*, pages 1–16. Springer, 2014.
46. Alessandro Solimando, Ernesto Jimenez-Ruiz, and Giovanna Guerrini. Minimizing conservativity violations in ontology alignments: Algorithms and evaluation. *Knowledge and Information Systems*, 2016.
47. Christian Strobl. *Encyclopedia of GIS*, chapter Dimensionally Extended Nine-Intersection Model (DE-9IM), pages 240–245. Springer, 2008.
48. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the Workshop on Evaluation of Ontology-based Tools (EON), Hiroshima (JP)*, 2004.
49. Élodie Thiéblin. Do competency questions for alignment help fostering complex correspondences? In *Proceedings of the EKAW Doctoral Consortium 2018*, 2018.
50. Élodie Thiéblin, Fabien Amarger, Ollivier Haemmerlé, Nathalie Hernandez, and Cássia Trojahn dos Santos. Rewriting SELECT SPARQL queries from 1: n complex correspondences. In *Proceedings of the 11th International Workshop on Ontology Matching*, pages 49–60, 2016.
51. Elodie Thiéblin, Michelle Cheatham, Cassia Trojahn, Ondrej Zamazal, and Lu Zhou. The First Version of the OAEI Complex Alignment Benchmark. In *Proceedings of the International Semantic Web Conference (Posters and Demos)*, 2018.
52. Ondřej Zamazal and Vojtěch Svátek. The ten-year ontofarm and its fertilization within the onto-sphere. *Web Semantics: Science, Services and Agents on the World Wide Web*, 43:46–53, 2017.
53. Lu Zhou, Michelle Cheatham, and Pascal Hitzler. Towards association rule-based complex ontology alignment. In *Proceedings of the 9th Joint International Semantic Technology Conference JIST 2019, Hangzhou, China, November 25*, in press, 2019.
54. Lu Zhou, Michelle Cheatham, Adila Krisnadhi, and Pascal Hitzler. A complex alignment benchmark: Geolink dataset. In *Proceedings of the 17th International Semantic Web Conference, Monterey (CA, USA)*, pages 273–288, 2018.
55. Lu Zhou, Michelle Cheatham, Adila Krisnadhi, and Pascal Hitzler. Geolink dataset: a complex alignment benchmark from real-world ontology. *Data Intelligence*, in press, 2019.

Jena, Lisboa, Milano, Heraklion, Mannheim,
Oslo, London, Berlin, Bonn, Linköping,
Trento, Toulouse, Prague, Manhattan
November 2019

AnyGraphMatcher Submission to the OAEI Knowledge Graph Challenge 2019*

Alexander Lütke[1]

University of Mannheim, Germany

Abstract. Matching objects between two different data bases typically relies on syntactic similarity measurements and the exhausting of ontology restrictions. As opposed to them, AnyGraphMatcher (AGM) introduces an additional source of information for semantically matching data – i.e. the creation of word embeddings. AGM's key idea wraps around a stable marriage for determining best matching data objects between two data bases. Results on the OAEI knowledge graph track however indicate the need for a more advanced blocking technique. Results show that word embeddinngs are to be seen a supportive feature for mapping rather than a key source of information.

Keywords: Ontology matching · Word embeddings · Semi-supervised machine learning.

1 Presentation of the system

1.1 State, purpose, general statement

In recent years, data has developed into a differentiator for business success. But organizations gathered data typically comes from various sources, which are mutually heterogeneous and inconsistent. Identity resolution is required to locate and integrate common pieces of information between these data sources. More precisely, data elements between those data bases have to be compared to each other and a decision on whether they describe the same real world concept must be made. Most prevalent techniques resolve around the comparison of syntactic elements, like titles, labels or descriptions. However, those techniques fail to preserve the actual semantic meaning of data objects. Consider for example the word Berlin, either describing Germany's capital or a cargo ship. Just from the title, the semantic meaning of the word "Berlin" cannot be determined. Recent breakthrough in linguistics research on the latent representation of words offers a promising opportunity [3]. The main notion wraps around the distributional hypothesis by Harris, stating that the meaning of a word is defined by its context. First, the hypothesis referred to linguistics only. But the adaptation in Paulheim's RDF2Vec approach [5] showcased, that the same concept is applicable to (semi-)structured databases. ALOD2VEC [4] and DOME [1] are two

systems, which already used the idea of word embeddings for data integration in the 2018 OAEI challenge. Compared to them, AnyGraphMatcher (AGM) is a novel concept specifically for identity resolution, which extends RDF2Vec with the use of semi-supervised machine learning and stable marriage.

1.2 Specific techniques used

The task of identity resolution is perceived as a binomial statement whether two given data elements describe the same real-world object. The final goal is such a binomial prediction for the entire Cartesian product of all elements of two databases A and B. Essentially, AnyGraphMatcher employs a five step process to get there:

1. Blocking
2. Graph walk
3. Word embedding model
4. Semi-supervised machine learning
5. Stable marriage

The Cartesian product can be extremely large depending on the size of the input datasets, representing a burden to runtime performance. However, most of the object pairs from the Cartesian product will not be matching, particularly if the input datasets are considered free of duplicates. So **blocking** is required. Instead of performing the following, computationally expensive predictions on the whole Cartesian product, a number of candidate-pairs is chosen based on an efficient similarity computation first. All pairs except the candidate-pairs are directly predicted non-matching. For the efficient candidate selection, a levenshtein automata is utilized, which measures *syntactic* distances. More precisely, edit distances up to two edits per word in a string are calculated. Note one major limitation in this concept: An assumption is met that actually corresponding data objects have similar labels. This might increase precision, but decrease recall.

Afterwards, a **graph walk** is employed, which iterates through the set of vertices and edges of an ontology graph. While walking through the graph, visited paths are written down, so that a corpus file is created in the end. The more detailed, recursive procedure looks as follows: For each vertice, an outgoing edge is selected. The edge is traced to its endpoint and the selection-procedure is started from there on again. At each vertice visited, such a selection is triggered n times. Furthermore, after excelling a distance of k steps from the vertice, where the path started, the procedure is terminated. Due to runtime limitations, for the OAEI submission, n equalling the number of outgoing edges of the current vertice and $k = 1$ was chosen. Basically, this boils down to a simple NT-file representation of an ontology. Further improvements could be derived from experiments with larger k values.

The graph walk has generated a text corpus, which can be passed to a **word embedding model** to form a latent representation of each word occurring in the

corpus. As the utilized SkipGram neural model is quite prevalent today, further details on the concept remains to the respective works in the literature. Since the SkipGram model is however highly configurable, most parameters have been adopted from the general recommendation of the inventors (e.g. hidden layer size = 100) Only the number of epochs has been modified. Due to runtime limitations, those depend on the size of the generated corpus, but halts between 10 and 250. After running the SkipGram model on the corpus, each resource in the input ontology can be assigned a 100-dimensional vector.

The final goal is a prediction whether two candidate pairs correspond to each other. This is achieved by **semi-supervised machine learning**. Supervision requires for a gold standard, which cannot be presupposed for said mapping tasks. That is why AnyGraphMatcher employs an automatic gold standard set generation. For this purpose, the candidate-pairs from the blocking are considered once again. Special attention is paid to the similarity of data object's labels. For the efficient calculation, the library "Apache Lucene" is used, which measures similarity in a value range of up to 2.5 for identical strings and 0.0 for very distant strings.[1] Very similar candidate-pairs with a similarity-score of 2.0 or above are assumed matches. All further candidate-pairs, in which one of the matching data objects occurs, are assumed non-matches. In the end, this gold standard captures most apparent matches, but differentiates them by the inclusion of matches (i.e. positives) and non-matches (i.e. negatives).

With the gold standard in place, a binary machine learning classifier – here an XGBoost – can work properly. Besides, the classifier takes more information than just the syntactic similarity into account. I.e. latent information is derived from the SkipGram model and passed to the classifier. Pairwise cosine similarity, Euclidean distance and SkipGram context probability (i.e. the probability that resource x appears in the context of resource y) are calculated. Other than expected, the binary classification is not meant to be the final prediction step however. Rather, it is implemented as another, more enhanced way of blocking. This is done by up-sampling the matching pairs during training until there are 1.5 times as many matches as non-matches. This procedure makes sure, that only *very likely* non-matches are classified negatives and excluded from further processing.

The final prediction is achieved by **stable marriage** under the assumption of 1:1 cardinality mappings. Here, each data object is considered in isolation first. All remaining candidate-pairs, in which a given data object appears, are extracted. For all found candidate-pairs, an overall similarity score is calculated. That score includes cosine similarity, Euclidean distance, SkipGram context probability and levenshtein distance. For each of these measures, the relative similarity in comparison with other candidate-pairs is computed. This is quantified by the position p in the following formula:

$$sim_{relative} = 2^{-(p-1)} \tag{1}$$

[1] The exact calculation is not meant to be explained here. For further details, refer to https://lucene.apache.org/core/3_5_0/scoring.html.

The following tables 1.2, 1.2 and 1.2 illustrate the idea once more. The focused data object in these tables is called A. A might match to B, C and D. Table 1.2 shows various thought similarity values between the candidate objects and A. Stable marriage goes on as follows to determine which one of the candidates is the best matching one: Table 1.2 indicates the order, how well a candidate-pair matches compared to other candidate pairs. Note that each of the similarity measure is still considered in isolation here. Table 1.2 then translates the ordering into values computed by the equation above. The final score in table 1.2 is calculated by summing the translated values for each of the candidate pairs. The one pair with the highest total score is assumed to match best. As a secondary criterion for further discrimination, the score derived from Levenshtein distance is taken. The way the final score is calculated is to be seen as an optimisable characteristic of AGM.

Candidate-pair	**Cos. sim**	**Eucl. dist.**	**Lev. dist.**	**P(Context)**
A – B	0.5	1.5	4	0.1
A – C	0.8	1.0	8	0.15
A – D	0.7	1.4	5	0.2

Table 1. Similarity measures of candidate pairs

Candidate-pair	**Relative Position**			
	Cos. sim	**Euclid. dist**	**Lev. dist.**	**P(Context)**
A – B	3	3	1	3
A – C	1	1	3	2
A – D	2	2	2	1

Table 2. Ordering candidate pairs based on their relative similarity

Candidate-pair	**Preliminary score**				**Total score**
	Cos. sim	**Euclid. dist**	**Lev. dist.**	**P(Context)**	
A – B	0.25	0.25	1	0.25	1.75
A – C	1	1	0.25	0.5	**2.75**
A – D	0.5	0.5	0.5	1	2.5

Table 3. Ordering candidate pairs based on a final score calculation

1.3 Summary of the system's limitations

Despite AnyGraphMatcher's exploitation of a wide range of characteristics of data objects, it suffers from two major limitations:

First, strong confidence is set to syntactic similarity. Basically two assumption lead to this conclusion: (1) data objects, which are syntactically similar, do match (see gold standard generation) and (2) actually matching data objects have similar labels (see blocking).

Second limitation is the recall-bias in the entire pipeline. Note that the only three steps in place to predict candidate-pairs negative are (1) blocking, (2) semi-supervised machine learning and (3) stable marriage. Blocking (1) and semi-supervised machine learning (2) themselves are recall-biased. So they prefer to predict (syntactically) similar samples as positives rather than negatives. Stable marriage can only predict negatives, if a data object has already been identified a match with another data object. So all in all, there is no strict exclusion of negatives from the set of candidate-pairs. This might raise precision, but reduce recall.

In sum, AGM can basically exploit semantics, if and only if the underlying data sets consistently follow the syntactic similarity assumption from above.

1.4 Adaptations made for the evaluation

For the OAEI submission, the melt framework provided by the University of Mannheim has been used [2]. Melt handles most of the regulations required for submitting matcher systems to the OAEI challenge. Since melt is originally written in Java, while AGM is mainly developed in Python, melt is used as a wrapper service, that calls the AGM pipeline by starting a new Python-process. Furthermore, the blocking process has been adapted to run more efficiently on the larger of the data sets in the OAEI knowledge graph track. A very strict blocking is applied, that initially excludes a lot candidate matches. Whether this technique harms recall is to be clarified in the results section.

1.5 Link to the system and parameters file

The implementation of AGM can be found on Github using the link https://github.com/XLexxaX/AnyGraphMatcher/tree/SUBMISSION.

2 Results

The following paragraphs shortly outline the results of AGM compared to the baseline figures. It therefore refers to the full result table available online on http://oaei.ontologymatching.org/2019/results/knowledgegraph/index.html. For the reason of comprehensibility, table 4 lists a more compact overview of the knowledge graph track.

System	Prec.	F-m.	Rec.
AGM	0.48 (0.48)	0.25 (0.25)	0.17 (0.17)
AML	0.72 (0.90)	0.70 (0.88)	0.69 (0.86)
baselineAltLabel	0.89 (0.89)	0.84 (0.84)	0.80 (0.80)
baselineLabel	0.95 (0.95)	0.81 (0.81)	0.71 (0.71)
DOME	0.74 (0.92)	0.70 (0.88)	0.67 (0.84)
FCAMap-KG	0.91 (0.91)	0.85 (0.85)	0.79 (0.79)
LogMapKG	0.40 (0.40)	0.54 (0.54)	0.84 (0.84)
LogMapLt	0.73 (0.91)	0.66 (0.83)	0.61 (0.76)
Wiktionary	0.91 (0.91)	0.80 (0.80)	0.71 (0.71)

Table 4. Comprehensive overview of the OAEI knowledge graph track results

2.1 Marvel Cinematic Universe Wiki ~ *MarvelDatabase*

With an overall F-score of 11%, AGM fails to output proper mappings on the first of the five mapping tasks. Taking a closer look at the five steps in the AGM pipeline, the exclusion of many candidate-mappings during blocking stands out.

2.2 Memory Alpha ~ *MemoryBeta*

The mapping of Memory Alpha to Memory Beta yielded slightly better results with an F-score of 32%. But still, the purely syntax-based baseline oupterforms AGM by approximately 50%. Notable is however, that this time, precision (47%) is significantly better than recall (24%).

2.3 Memory Alpha ~ *StarTrekExpandedUniverse*

The observations from Memory Alpha and Memory Beta continue throughout the remaining three mapping tasks. All in all, an F-score of 30% was achieved when mapping Memory Alpha to Star Trek Expanded Universe. The baseline of 91% F-score is missed.

2.4 Star Wars Wiki ~ *StarWarsGalaxiesWiki*

For the Star wars wiki mapping, again 30% F-score is achieved. The baseline F-score of 67% is out of reach. Note however the even larger gap of 52% between AGM's precision and recall this time.

2.5 Star Wars Wiki ~ *TheOldRepublicWiki*

Repeatedly, a 52% gap between recall and precision is conspicious. The F-score of 20% does not meet the pretension of the baseline by far.

3 General comments

3.1 Comments on the results

In sum, results of AGM on the knowledge graph track are relatively weak compared to the baseline figures. Mainly recall is lacking behind the competition's results. This can be traced back to the strict blocking technique. However, precision also lacks behind the baseline figures. Recap, that a levenshtein automata has been used, which can only measure edit distances of up to 2 edits per word. In case two long texts are compared, this restriction leads to imprecise measuring. So the levenshtein automata as implemented in AGM is rather an approximation of syntactic similarity. Nevertheless, a static threshold has been used for blocking (see section 1.2), such that in the end precision suffers as well.

3.2 Discussions on the way to improve the proposed system

In order to compensate for the weak results, another way has to be found block based on a weaker threshold, while ensuring runtime efficiency of the AGM pipeline. One idea is to loosen the current threshold and introduce a second blocking step, that blocks based on exact edit distances for all candidates found by the levensthein automata.

4 Conclusion

AGM follows a novel approach to data mappings by utilizing the idea of word embeddings. It implements a five step process including blocking, a graph walk, embedding creation, semi-supervised machine learning and stable marriage. By combining different similarity measures derived from syntax and word embeddings, AGM aims to yield semantically correct mappings. However results show a relatively poor performance compared to the purely syntax based baseline figures. A strict and imprecise blocking technique has been identified a root cause. Though the results cannot achieve the baseline figures, they provide a valuable outcome for AGM's approach in general: The stable marriage depends a lot on the upstream steps and suffers from error-propagation. This implies that features derived from embeddings cannot be solely used for mapping. Embeddings are to be seen an approximation of concept's semantic meaning, such that they can additionally support in distinguishing them. In order to compensate for this observation in the future, a more advanced blocking technique is required.

References

1. DOME results for OAEI 2018. In *OM@ISWC*, volume 2288 of *CEUR Workshop Proceedings*, pages 144–151, Karlsruhe, 2018. CEUR-WS.org.
2. Sven Hertling, Jan Portisch, and Heiko Paulheim. MELT - Matching EvaLuation Toolkit. In *Semantics 2019 SEM2019 Proceedings*, Karlsruhe, 2019, to appear.

3. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.
4. Jan Portisch and Heiko Paulheim. Alod2vec matcher. In *OM@ISWC*, volume 2288 of *CEUR Workshop Proceedings*, pages 132–137. CEUR-WS.org, 2018.
5. Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *The Semantic Web - ISWC 2016 : 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, volume 9981, pages 498–514, Cham, 2016. Springer International Publishing.

ALIN Results for OAEI 2019

Jomar da Silva[1], Carla Delgado[1], Kate Revoredo[1], and Fernanda Araujo Baião[2]

[1] Graduate Program in Informatics, Federal University of Rio de Janeiro (UFRJ), Brazil
[2] Department of Industrial Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil
jomar.silva@uniriotec.br,carla@ppgi.ufrj.br,katerevoredo@ppgi.ufrj.br, fbaiao@puc-rio.br

Abstract. [1] ALIN is an ontology matching system specialized in the interactive ontology matching, and its main characteristic is the use of expert feedback to improve the set of selected mappings, using semantic and structural techniques to make this improvement. This paper describes its configuration for the OAEI 2019 competition and discusses its results.

Keywords: ontology matching, Wordnet, interactive ontology matching, ontology alignment, interactive ontology alignment

1 Presentation of the System

Due to the advances in information and communication technologies, a large amount of data repositories became available. Those repositories, however, are highly semantically heterogeneous, which hinders their integration. Ontology matching has been successfully applied to solve this problem, by discovering mappings between two distinct ontologies which, in turn, conceptually define the data stored in each repository. Among the various ontology matching approaches that exist in the literature, interactive ontology matching includes the participation of domain experts to improve the quality of the final alignment [1]. ALIN is an interactive ontology matching system and has been participating in all OAEI editions since 2016, with improving results.

1.1 State, Purpose and General Statement

ALIN is a system for interactive ontology matching that consists of two steps: one non-interactive and one interactive. In the first step, ALIN chooses the first mappings, among which some are directly placed in the alignment and others are presented to the expert. In the 2019 version, ALIN uses new techniques to improve the first step, thus placing more mappings directly in the alignment without having to present them to the expert.

1.2 Specific Techniques Used

ALIN handles three sets of mappings: (i) Accepted, which is a set of mappings definitely to be retained in the alignment; (ii) Selected, which is a set of mappings where each is yet to be decided if it will be included in the alignment; and (iii) Suspended, which is a set of mappings that have been previously selected, but (temporarily or permanently) filtered out of the alignment.

Given the previous definitions, ALIN procedure follows 5 Steps, described as follows:

1. Select mappings: select the first mappings and automatically accepts some of them. We explain the selection and acceptance process below;
2. Filter mappings: suspend some selected mappings, using lexical criteria for that;
3. Ask expert: accepts or rejects selected mappings, according to expert feedback
4. Propagate: select new mappings, reject some selected mappings or unsuspend some suspended mappings (depending on newly accepted mappings)
5. Go back to 3 as long as there are undecided selected mappings

All versions of ALIN (since its very first OAEI participation) follow this general procedure. In this 2019 version, however, we introduced modifications in Step 1. In previous versions, ALIN automatically accepted only the entities with the same name. In this version, ALIN also automatically accepts the entities whose names are synonyms or with variations in name words. ALIN searches synonyms in the Wordnet. In the Anatomy track, ALIN uses the FMA ontology too.

ALIN applies the following techniques:

- Line 1. ALIN selects mappings using linguistic similarities between entity names. ALIN uses synonyms and variations in entity name words to automatically accept mappings. At this time, ALIN automatically selects and accepts only concept mappings. To do that, ALIN uses linguistic metrics. ALIN uses the Wordnet and domain-specific ontologies (the FMA Ontology in the Anatomy track) to find synonyms between entities.
- Line 2. ALIN suspends the selected mappings whose entities have low lexical similarity. We use the Jaccard, Jaro-Wrinkler, and n-gram lexical metrics to calculate the lexical similarity of the selected mappings. We based the process of choosing the similarity metrics used by ALIN on the result of these metrics in assessments [2]. It is important to know that these suspended mappings can be unsuspended later, by structural analysis, as proposed in [3].
- Line 3. At this point, the expert interaction begins. ALIN sorts the selected mappings in a descending order according to the sum of similarity metric values. The sorted selected mappings are submitted to the expert.
- Line 4. Initially, the set of selected mappings contains only concept mappings. At each interaction with the expert, if the expert accepts the mapping, ALIN (i) removes from the set of selected mappings all the mappings that

compose the mapping anti-pattern [4][5] (we explain mapping anti-pattern below) with the accepted mappings; (ii) selects data property (like [6]) and object property mappings related to the accepted concept mappings; (iii) unsuspends all concept mappings whose both entities are subconcepts of the concept of an accepted mapping, following a similar technique proposed in our previous work [3].

– Line 5. The interaction phase continues until there are no selected mappings.

An ontology may have construction constraints, such as a concept cannot be equivalent to its superconcept. An alignment may have other constraints like, for example, an entity of ontology O cannot be equivalent to two entities of the ontology O'. A mapping anti-pattern is a combination of mappings that generates a problematic alignment, i.e., a logical inconsistency or a violated constraint.

1.3 Link to the System and Parameters File

ALIN is available [2] as a package to be run through the SEALS client.

2 Results

Interactive ontology matching is the focus of the ALIN system. Comparing its results in the 2019 campaign to its previous participations (Table 5), ALIN improvements include an expressive reduction on the number of interactions with the expert and the increase of the quality of the generated alignment.

2.1 Comments on the Participation of the ALIN in Non-Interactive Tracks

ALIN used new techniques to automatically accept mappings. These techniques led to an increase in the F-Measure of non-interactively generated alignment, which shows the effectiveness of the techniques. (Table 1 and Table 2). Conference track, unlike the Anatomy track, has relationship mappings and attribute mappings that ALIN does not automatically accept, thus making the F-Measure on the Conference track, although higher than last year, still low.

Table 1. Participation of ALIN in Anatomy Non-Interactive Track - OAEI 2018[7]/2019[8]

Year	Precision	Recall	F-measure
2018	0.998	0.611	0.758
2019	0.974	0.698	0.813

[2] $https://drive.google.com/file/d/1SxJL6fLRVqI84epm8DbA_MlcscEoGbgZ/view?usp=sharing$

Table 2. Participation of ALIN in Conference Non-Interactive Track - OAEI 2018/2019[9]

Year	Precision	Recall	F-measure
2018	0.81	0.42	0.55
2019	0.82	0.43	0.56

2.2 Comments on the Participation of the ALIN in Interactive Tracks

In the Anatomy track, ALIN was tied for second in quality (F-Measure) with slightly lower total requests (Table 3). In the Conference track, ALIN was tied for first in quality with a slightly higher total request (Table 4).

Table 3. Participation of ALIN in Anatomy Interactive Track - Error Rate 0.0[10]

Tool	Precision	Recall	F-measure	Total Requests
ALIN	0.979	0.85	0.91	365
AML	0.968	0.948	0.958	236
LogMap	0.982	0.846	0.909	388

Table 4. Participation of ALIN in Conference Interactive Track - Error Rate 0.0[10]

Tool	Precision	Recall	F-measure	Total Requests
ALIN	0.914	0.695	0.79	228
AML	0.91	0.698	0.79	221
LogMap	0.886	0.61	0.723	82

Interactive Anatomy Track In this track, ALIN has had a decrease in the number of expert interactions and an increase in the quality of the generated alignment, showing that the new techniques used to automatically accept correct mappings are effective (Table 5).

ALIN used the FMA ontology to help find synonyms between the two ontologies of the Anatomy track. The Foundational Model of Anatomy Ontology (FMA) is a reference ontology for the domain of Human anatomy [3].

[3] "Foundational Model of Anatomy Ontology". Available at http://sig.biostr.washington.edu/projects/fm/AboutFM.html Last accessed on Oct, 11, 2019.

Interactive Conference Track In this track, ALIN has had a decrease in the number of expert interactions keeping a good quality of the generated alignment (Table 7).

2.3 Comparison of the Participation of ALIN in OAEI 2019 with his Participation in OAEI 2018

In this version, ALIN uses new techniques to automatically accept mappings. These techniques use synonyms and word variations to find equal entities between the two ontologies. ALIN also started to use FMA ontology as an external resource.

The use of the new techniques proved to be effective as it reduced the number of interactions while keeping a good level of quality. The new techniques also increased the quality of the alignment generated in Anatomy interactive tracking, where ALIN used the FMA ontology.

It is not always possible to use an external resource to find synonyms between entities of two ontologies, but when this is possible, the results showed that it is worth it.

The quality of the alignment generated by ALIN is dependent on the correct expert feedback, as expert responses are used to select new mappings. When ALIN selects wrong mappings, the quality of the generated alignment tends to decrease. But if we compare this year's quality decline with last year's, we see that this fall is less sharp (Table 6 and Table 8). The less sharp decline in quality is because we need less user interaction as we are automatically accepting more mappings.

The organization of FMA ontology in memory and the search for synonyms and word variations led to longer run time (Table 9 and Table 10)

Table 5. Participation of ALIN in Anatomy Interactive Track - OAEI 2016[11]/2017[12]/2018[7]/2019[10] - Error Rate 0.0

Year	Precision	Recall	F-measure	Total Requests
2016	0.993	0.749	0.854	803
2017	0.993	0.794	0.882	939
2018	0.994	0.826	0.902	602
2019	0.979	0.85	0.91	365

3 General Comments

Evaluating the results, we can see that the system has improved, although it can improve even further, towards:

– handling user error rate;

Table 6. F-Measure of ALIN in Anatomy Interactive Track - OAEI /2018[7]/2019[10] - with Different Error Rates

Year	Error rate 0.0	Error rate 0.1
2018	0.902	0.854
2019	0.91	0.889

Table 7. Participation of ALIN in Conference Interactive Track - OAEI 2016[11]/2017[12]/2018[7]/2019[10] - Error Rate 0.0

Year	Precision	Recall	F-measure	Total Requests
2016	0.957	0.735	0.831	326
2017	0.957	0.731	0.829	329
2018	0.921	0.721	0.809	276
2019	0.914	0.695	0.79	228

Table 8. F-Measure of ALIN in Conference Interactive Track - OAEI /2018[7]/2019[10] - with Different Error Rates

Year	Error rate 0.0	Error rate 0.1
2018	0.809	0.705
2019	0.79	0.725

Table 9. Run Time (sec) in Anatomy Interactive Track - OAEI /2018[13]//2019[10]

Tool	2018	2019
ALIN	317	2132
AML	48	82
LogMap	23	29

Table 10. Run Time (sec) in Conference interactive track - OAEI /2018[13]/2019[10]

Tool	2018	2019
ALIN	106	397
AML	35	34
LogMap	37	37

– generating a higher quality initial alignment in its non-interactive phase;
– reducing the number of interactions with the expert;

And there was a worsening run time, where we could improve too.

3.1 Conclusions

ALIN used new techniques to automatically accept new mappings. They have been effective in reducing the number of interactions, while also keeping good quality in the generated alignment. In the case of the Anatomy track, these new techniques both decreased the number of interactions and increased the quality of the generated alignment. We can explain this quality improvement in this track by the use of the FMA ontology as a new external resource. With the use of the new techniques in both Anatomy and Conference tracks, there has been a less sharp drop in quality as the expert makes mistakes. Nevertheless, ALIN had an increase in run time due to the use of the new techniques, which may be addressed in future work.

References

1. Paulheim, H., Hertling, S., Ritze, D.: Towards Evaluating Interactive Ontology Matching Tools. Lecture Notes in Computer Science **7882** (2013) 31–45
2. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: Proceedings of the 12th International Semantic Web Conference - Part II. ISWC '13, New York, NY, USA, Springer-Verlag New York, Inc. (2013) 294–309
3. Silva, J., Baião, F., Revoredo, K., Euzenat, J.: Semantic interactive ontology matching: Synergistic combination of techniques to improve the set of candidate correspondences. In: OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching. Volume 2032. (2017) 13–24
4. Guedes, A., Baião, F., Shivaprabhu, Revoredo, R.: On the Identification and Representation of Ontology Correspondence Antipatterns. In: Proc. 5th Int. Conf. Ontol. Semant. Web Patterns (WOP'14), CEUR Work. Proc. (2014)
5. Guedes, A., Baião, F., Revoredo, K.: Digging Ontology Correspondence Antipatterns. In: Proceeding WOP'14 Proc. 5th Int. Conf. Ontol. Semant. Web Patterns. Volume 1032. (2014) 38—-48
6. Silva, J., Revoredo, K., Baião, F.A., Euzenat, J.: Interactive Ontology Matching: Using Expert Feedback to Select Attribute Mappings. (2018)
7. Silva, J., Baião, F., Revoredo, K.: Alin results for oaei 2018. In: Ontology Matching: OM-2018: Proceedings of the ISWC Workshop. OM'18 (2018) 117–124
8. : Results for oaei 2019 - anatomy track. http://oaei.ontologymatching.org/2019/results/anatomy/ Accessed: 2019-10-11.
9. : Results of evaluation for the conference track within oaei 2019. http://oaei.ontologymatching.org/2019/results/conference/index.html Accessed: 2019-10-11.
10. : Results for oaei 2019 - interactive track. http://oaei.ontologymatching.org/2019/results/interactive/ Accessed: 2019-10-11.
11. Silva, J., Baião, F., Revoredo, K.: Alin results for oaei 2016. In: OM-2016: Proceedings of the Eleventh International Workshop on Ontology Matching. OM'16 (2016) 130–137
12. Silva, J., Baião, F., Revoredo, K.: Alin results for oaei 2017. In: OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching. OM'17 (2017) 114–121
13. : Results for oaei 2018 - interactive track. http://oaei.ontologymatching.org/2018/results/interactive/index.html Accessed: 2019-10-11.

AML and AMLC Results for OAEI 2019

Daniel Faria[1], Catia Pesquita[2], Teemu Tervo[2]
Francisco M. Couto[2], and Isabel F. Cruz[3]

[1] BioData.pt & INESC-ID, Lisboa, Portugal
[2] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
[3] ADVIS Lab, Department of Computer Science, University of Illinois at Chicago, USA

Abstract. AgreementMakerLight (AML) is an ontology matching system designed with scalability, extensibility and satisfiability as its primary guidelines, as well as an emphasis on the ability to incorporate external knowledge. In OAEI 2019, AML's development focused mainly on expanding its range of complex matching algorithms, but there were also improvements on its instance matching pipeline and ontology parsing algorithm. AML remains the system with the broadest coverage of OAEI tracks, and among the top performing systems overall.

1 Presentation of the System

1.1 State, Purpose, General Statement

AgreementMakerLight (AML) is an ontology matching system inspired on AgreementMaker [1, 2] and drawing on its design principles, but with an added focus on scalability to tackle large ontology matching problems [8]. While initially focused primarily on the biomedical domain, it is currently a general purpose ontology matching system that is able to successfully tackle a broad range of problems.

AML is primarily based on lexical matching algorithms [9], but also includes structural algorithms for both matching and filtering, as well as its own logical repair algorithm [10]. It makes use of external biomedical ontologies and the WordNet as sources of background knowledge [7].

This year, our development of AML was mainly focused on expanding the arsenal of complex matching algorithms of AML to improve its performance in the new Complex Matching track. The complex matching version of AML, dubbed AMLC, remains separate from the main AML submission, as we have been as of yet unable to integrate the complex code into the main code-base.

In addition to these two versions, we again participated in the SPIMBENCH and Link Discovery tracks via the HOBBIT platform. In the case of SPIMBENCH, we participated with the HOBBIT adaptation of the main AML code-base. In the case of Link Discovery, we participated with two specialized versions of AML (AML-Spatial and AML-Linking for the Spatial and Linking tasks respectively) as had been the case in

OAEI 2017 and 2018, due to the unique characteristics of these matching tasks and to the unavailability of the TBox assertions in the HOBBIT datasets.

1.2 Specific Techniques Used

This section describes only the features of AML that are new for the OAEI 2019. It also describes AMLC, a variant of AML tailored to complex matching. For further information on AML's simple matching strategy, please consult AML's original paper [8] as well as the AML OAEI results publications of the last four editions [4, 5, 3, 6].

1.2.1 AML

Ontology Parsing

We made a few extensions to AML's ontology parser to enable it to infer the types of ontology properties declared only as rdf:property (which the OWL API interprets as annotation properties by default). There were critical to correctly interpret and match the datasets for the Knowledge Graph track.

Instance Matching

We refined AML's instance matching pipeline to more adequately distinguish between cases where lexical matching should be the primary strategy complemented by property-based matching, and cases where property-based matching should be the primary strategy, by using the ratios of labels per instances and property values per instances as deciding factors. These improvements were critical to AML's effectiveness on the Knowledge Graph track.

1.2.2 AMLC

For the complex matching track, we developed algorithms to tackle additional types of EDOAL mappings, namely mappings involving union class constructs. Furthermore, we refined the Attribute Occurrence Restrictions and Attribute Domain Restrictions algorithms developed last year to take into account instance data when available.

These changes allowed AML to match ontologies from the GeoLink dataset, in addition to those from the Conference dataset.

1.3 Adaptations made for the evaluation

As was the case last year, the Link Discovery submissions of AML are adapted to these particular tasks and datasets, as their specificities (namely the absence of a Tbox) demand a dedicated submission. The same is also true to some extent of AML's Complex Matching submission.
As usual, our submission included precomputed dictionaries with translations, to circumvent Microsoft® Translator's query limit.

1.4 Link to the system and parameters file

AML is an open source ontology matching system and is available through GitHub: https://github.com/AgreementMakerLight.

2 Results

2.1 Anatomy

AML's result was the same as in previous years, with 95% precision, 93.6% recall, 94.3% F-measure, and 83.2% recall++. It remains the best ranking system in this track by both F-measure and recall++.

2.2 Conference

AML's result was exactly the same as in recent years, with 74% F-measure according to the full reference alignment 1, 70% F-measure according to the extended reference alignment 2, 78% F-measure according to the discrete uncertain reference alignment, and 77% according to the continuous one. It remains the best ranking system in this track or tied for best by F-measure according to 4 of the 5 sets of reference alignments available. It ranks second by F-measure on the violation free version of reference alignment 2, as enforcing the removal of conservativity violations can produce undesired practical effects that are not aligned with AML's guiding principles, so our repair algorithm does not take them into account.

2.3 Multifarm

AML's results were similar to last year, ranking first with 45% F-measure in the different ontologies modality, but second with only 27% F-measure in the same ontologies modality. We are still unsure as to why AML performs worse in the same ontologies modality.

2.4 Complex Matching

AMLC was configured only for the Conference and Geolink datasets. It also produced results in the Hydrography dataset, but these were expectedly mediocre.
On the conference dataset, AMLC was the only system to participate in the non-populated version (using the simple reference alignment as input). It improved its recall in relation to last year (37% versus 25%) but this came at the expense of precision and so resulted in an identical F-measure of 34%. On the populated version, it had the highest range of coverage (query F-measure) with 46-50%.
On the GeoLink dataset, AMLC obtained a comparably modest F-measure of 32% (the top system had 60%).

2.5 Interactive Matching

AML had an identical performance to last year, as no changes were made to its interactive algorithms. It remains the system with the best F-measure in both the Anatomy and Conference datasets across all error rates (though it also has the best non-interactive F-measure in these datasets).

2.6 Large Biomedical Ontologies

AML had an F-measure of 93.3% in FMA-NCI small, 84.1% in FMA-NCI whole, 83.5% in FMA-SNOMED small, 69.7% in FMA-SNOMED whole, 81.8% in SNOMED-NCI small and 76.5% in SNOMED-NCI whole. In comparison with last year, its performance decreased in all large tasks, due to the erroneous addition of an imprecise matching algorithm in the matching pipeline when testing new configurations. Despite this, it remains the best performing system in five of the six tasks.

2.7 Disease and Phenotype

AML generated 2029 mappings in the HP-MP task, 330 of which were unique. It ranked third by F-measure according to the 3-vote silver standard, but this does not necessarily reflect its actual performance, as the unique mappings were not evaluated. If half of AML's unique mappings were proven correct, which is highly likely given the high precision AML obtains in other biomedical tasks, it would rank first in F-measure. In the DOID-ORDO task, it generated by far the most mappings (4781) and the most unique mappings (2342), and as a result had a relatively low F-measure according to the 3-vote silver standard (65.1%). Again, assessing the correctness of the unique mappings would be essential to gauge AML's true performance.

2.8 Biodiversity and Ecology

AML obtained the highest F-measure in both datasets, with 78.8% in the FLOPO-PTO task and 80.8% in the ENVO-SWEET task. It ranked first in recall and produced both the most mappings and the most unique mappings.

2.9 SPIMBENCH

AML obtained the same results as last year, with an F-measure of 86%, ranking third by F-measure.

2.10 Link Discovery

As in previous years, AML produced a perfect result (100% F-measure) in the Linking and all the Spatial tasks. It was among the most efficient systems in the later, and the only system participating in the former.

2.11 Knowledge Graph

AML was able to complete only four of the five tasks due to an unforeseen timeout in the largest task (which it had been able to carry out in testing). It produced an average F-measure of only 70% if the missing task is counted as zero, but of 88% when it is ignored. In fact, it ranked either first or second in F-measure in all the four tasks it completed.

3 General comments

3.1 Comments on the results

This year, AML was again the system that tackled the most OAEI tracks and datasets, maintaining its status as one of the broadest and best performing matching systems available to the community.
However, unlike AML's performance in traditional (simple) matching tracks, there is clearly room for improvement for AML in complex matching, as it had modest F-measures. We will strive to refine and improve AML's complex matching pipeline and contribute to the development of this branch of ontology matching.

3.2 Comments on the OAEI test cases

We once again laud the efforts of the organizers of both returning and especially new tracks, as the effort involved in organizing them cannot be overstated.
Nevertheless, we must again comment on the unsatisfactory evaluation in the Disease and Phenotype track by means of silver standards generated from the alignments produced by the participating systems via voting. We understand the effort required to build a manually curated reference alignment, but we believe that it is paramount to invest in it, in order to enable a proper evaluation of matching systems.

4 Conclusion

Like in recent years, AML was the matching system that participated in the most OAEI tracks and datasets, and it was among the top performing systems in most of them. AML's performance did not improve in any of the long-standing OAEI tracks, as most of our development effort went into tackling new challenges and extending the range of AML. We improved substantially our results in the knowledge graph track in comparison with last year, thanks to the extensions to AML's ontology parsing algorithm and its instance matching pipeline. We were also able to extend AML's complex matching algorithm portfolio, but despite this, AML complex matching performance requires further improvement. We will continue to invest in addressing this aspect of ontology matching in the near future

Acknowledgments

DF was funded by the EC H2020 grant 676559 ELIXIR-EXCELERATE and the Portuguese FCT Grant 22231 BioData.pt (co-financed by FEDER). CP and FMC were funded by the Portuguese FCT through the LASIGE Research Unit (UID/CEC/00408/2019). FMC was also funded by PTDC/CCI-BIO/28685/2017. CP was also funded by FCT (PTDC/EEI-ESS/4633/2014). The research of IFC and BSB was partially funded by NSF awards CCF-1934915, CNS-1646395, III-1618126, CCF-1331800, and III-1213013, and by NIGMS-NIH award R01GM125943.

References

1. I. F. Cruz, F. Palandri Antonelli, and C. Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.
2. I. F. Cruz, C. Stroe, F. Caimi, A. Fabiani, C. Pesquita, F. M. Couto, and M. Palmonari. Using AgreementMaker to Align Ontologies for OAEI 2011. In *ISWC International Workshop on Ontology Matching (OM)*, volume 814 of *CEUR Workshop Proceedings*, pages 114–121, 2011.
3. D. Faria, B. S. Balasubramani, V. R. Shivaprabhu, I. Mott, C. Pesquita, F. M. Couto, and I. F. Cruz. Results of AML in OAEI 2017. In *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, page 122, 2017.
4. D. Faria, C. Martins, A. Nanavaty, D. Oliveira, B. S. Balasubramani, A. Taheri, C. Pesquita, F. M. Couto, and I. F. Cruz. AML results for OAEI 2015. In *Ontology Matching Workshop*. CEUR, 2015.
5. D. Faria, C. Pesquita, B. S. Balasubramani, C. Martins, J. Cardoso, H. Curado, F. M. Couto, and I. F. Cruz. OAEI 2016 results of AML. In *Ontology Matching Workshop*. CEUR, 2016.
6. D. Faria, C. Pesquita, B. S. Balasubramani, T. Tervo, D. Carriço, R. Garrilha, F. M. Couto, and I. F. Cruz. Results of aml participation in oaei 2018. In *OM-2018: Proceedings of the Thirteenth International Workshop on Ontology Matching*, pages 125–131, 2018.
7. D. Faria, C. Pesquita, E. Santos, I. F. Cruz, and F. M. Couto. Automatic Background Knowledge Selection for Matching Biomedical Ontologies. *PLoS One*, 9(11):e111226, 2014.
8. D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto. The AgreementMakerLight Ontology Matching System. In *OTM Conferences - ODBASE*, pages 527–541, 2013.
9. C. Pesquita, D. Faria, C. Stroe, E. Santos, I. F. Cruz, and F. M. Couto. What's in a "nym"? Synonyms in Biomedical Ontology Matching. In *International Semantic Web Conference (ISWC)*, pages 526–541, 2013.
10. E. Santos, D. Faria, C. Pesquita, and F. M. Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PLoS ONE*, 10(12):e0144807, 2015.

AROA Results for 2019 OAEI*

Lu Zhou[1], Michelle Cheatham[2], and Pascal Hitzler[1]

[1] DaSe Lab, Kansas State University, Manhattan KS 66506, USA
{luzhou, hitzler}@ksu.edu
[2] Wright State University, Dayton OH 45435, USA
michelle.cheatham@wright.edu

Abstract. This paper introduces the results of alignment system AROA in the OAEI 2019 campaign. AROA stands for Association Rule-based Ontology Alignment system. This ontology alignment system can produce simple and complex alignment between ontologies that share common instance data. This is the first participation of AROA in the OAEI campaign, and it produces best performance on one of complex benchmarks (GeoLink).

1 Presentation of the system

1.1 State, purpose, general statement

AROA (Association Rule-based Ontology Alignment) system is aimed to automatically generate simple and complex alignment between two and more ontologies. These ontologies would be required to share common instance data because AROA relies on association rule mining and would require these instances as inputs. After generating a set of association rules, AROA utilizes some simple and complex correspondences that have been widely accepted in Ontology Matching community [4, 6] to further narrow the large number of rules down to more meaningful ones and finally establishes the alignments.

1.2 Specific techniques used

Figure 1 illustrates the overview of AROA alignment system. In this section, we introduce each step of AROA alignment system along with some concepts that we frequently use in AROA system, such as association rule mining, FP-growth algorithm, and complex alignment generation.

Clean Triple. First, AROA extracts all triples as the format of ⟨Subject, Predicate, Object⟩ from the source and target ontologies. Each item in a triple is expressed as a web URI. After collecting all of the triples, we clean the data based on the following criteria: we only keep the triples that contain at least one entity under the source or the target ontology namespace or the triples contain rdf:type information, since our algorithm relies on this information.

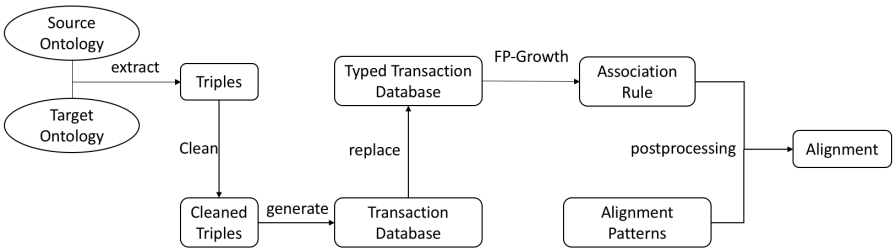


Fig. 1. Overview of AROA Alignment System

Generate Transaction Database. After filtering process, we generate the transaction database as the input for the FP-growth algorithm. Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of distinct attributes called items. Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions where each transaction in D has a unique transaction ID and contains a subset of the items in I. Table 1 shows a list of transactions corresponding to a list of triples. Instance data can be displayed as a set of triples, each consisting of subject, predicate, and object. Here, subjects represent the identifiers and the set of corresponding properties with the objects represent transactions, which are separated by the symbol "|". I.e., a transaction is a set $T = (s, Z)$ such that s is a subject, and each member of Z is a pair (p, o) of a property and an object such that (s, p, o) is an instance triple.

Generate Typed Transaction Database. Then we replace the object in the triples with its rdf:type[3] because we focus on generating schema-level (rather than instance-level) mapping rules between two ontologies, and the type

[3]If there are multiple types of the object, it can also combine the subject and predicate as additional information to determine the correct type, or keep both types as two triples.

Table 1. Triples and Corresponding Transactions

s_1	p_1	o_1
s_1	p_2	o_2
s_1	p_4	o_4
s_2	p_1	o_1
s_2	p_2	o_2
s_2	p_3	o_3
s_2	p_4	o_4
s_3	p_1	o_1
s_3	p_2	o_2

TID	Itemsets
s_1	$p_1\|o_1$, $p_2\|o_2$, $p_4\|o_4$
s_2	$p_1\|o_1$, $p_2\|o_2$, $p_3\|o_3$, $p_4\|o_4$
s_3	$p_1\|o_1$, $p_2\|o_2$

Table 2. Original Transaction Database

TID	Itemsets
x_1	gbo:hasAward|y_1, gmo:fundedBy|y_2
x_2	gbo:hasFullName|y_3, gmo:hasPersonName|y_4
x_3	rdf:type|gbo:Cruise, rdf:type|gmo:Cruise

Table 3. Typed Transaction Database

TID	Itemsets
x_1	gbo:hasAward|gbo:Award, gmo:fundedBy|gmo:FundingAward
x_2	gbo:hasFullName|xsd:string, gmo:hasPersonName|gmo:PersonName
x_3	rdf:type|gbo:Cruise, rdf:type|gmo:Cruise

information of the object is more meaningful than the original URI. If an object in a triple has rdf:type of a class in the ontology, we replace the URI of the object with its class. If the object is a data value, the URI of the object is replaced with the datatype. If the object already is a class in the ontology, it remains unchanged. Tables 2 and 3 show some examples of the conversion.

Generate Association Rules. Our alignment system mainly depends on a data mining algorithm called association rule mining, which is a rule-based machine learning method for discovering interesting relations between variables in large databases [3]. Many algorithms for generating association rules have been proposed, like Apriori [1] and FP-growth algorithm [2]. In this paper, we use FP-growth to generate association rules between ontologies, since the FP-growth algorithm has been proven superior to other algorithms [2]. The FP-growth algorithm is run on the transaction database in order to determine which combinations of items co-occur frequently. The algorithm first counts the number of occurrences of all individual items in the database. Next, it builds an FP-tree structure by inserting these instances. Items in each instance are sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items in each instance that do not meet the predefined thresholds, such as minimum support and minimum confidence (see below for these terms), are discarded. Once all large itemsets have been found, the association rule creation begins. Every association rule is composed of two sides. The left-hand-side is called the antecedent, and the right-hand-side is the consequent. These rules indicate that whenever the antecedent is present, the consequent is

Table 4. Examples of Association Rules

Antecedent	Consequent
$p_4\|o_4$, $p_1\|o_1$	$p_2\|o_2$
$p_2\|o_2$	$p_1\|o_1$
$p_4\|o_4$	$p_1\|o_1$

Table 5. The Alignment Pattern Types Covered in AROA System

Pattern	Category
Class Equivalence	1:1
Class Subsumption	1:1
Property Equivalence	1:1
Property Subsumption	1:1
Class by Attribute Type	1:n
Class by Attribute Value	1:n
Property Typecasting Equivalence	1:n
Property Typecasting Subsumption	1:n
Typed Property Chain Equivalence	m:n
Typed Property Chain Subsumption	m:n

likely to be as well. Table 4 shows some examples of association rules generated from the transaction database in Table 1.

Generate Alignment. AROA utilizes some simple and complex correspondences that have been widely accepted in Ontology Matching community to further filter rules [4, 6] and finally generate the alignments. There are totally 10 different types of correspondences that AROA covers in this year. Table 5 lists all the simple and complex alignment correspondences and corresponding category. Since the association rule mining might generate a large number of rules, in order to narrow the association rules down to a smaller set, AROA follows these patterns to generate corresponding alignments. For example, Class by Attribute Type (CAT) is a classic complex alignment pattern. This type of pattern was first introduced in [4]. It states that a class in the source ontology is in some relationship to a complex construction in the target ontology. This complex construction may comprise an object property and its range. Class C_1 is from ontology O_1, and object property op_1 and its range t_1 are from ontology O_2.

Association Rule format: $\mathsf{rdf{:}type}|C_1 \rightarrow \mathsf{op_1}|\mathsf{t_1}$
Example: $\mathsf{rdf{:}type}|\mathsf{gbo{:}PortCall} \rightarrow \mathsf{gmo{:}atPort}|\mathsf{gmo{:}Place}$
Generated Alignment: $\mathsf{gbo{:}PortCall(x)} \rightarrow \mathsf{gmo{:}atPort(x, y)} \wedge \mathsf{gmo{:}Place(y)}$

In this example, this association rule implies that if the subject x is an individual of class gbo:PortCall, then x is subsumed by the domain of gmo:atPort with its range gmo:Place. The equivalence relationship can be generated by combining another association rule holding the reverse information. Other simple and complex alignments are also generated by following the same steps.

1.3 Adaptations made for the evaluation

AROA is an instance-based ontology alignment system. Therefore, AROA embeds Apache Jena Fuseki server in the system. The ontologies are first downloaded from the SEALS repository. And then, AROA uploads and stores the ontologies in the embedded Fuseki server, which might take some time for this

Table 6. The Number of Alignments Found on GeoLink Benchmark

Alignment Patterns	Category	Reference Alignment	AROA	
-	-	-	# of Correct Entities	# of Correct Relation
Class Equiv.	1:1	10	10	10
Class Subsum.	1:1	2	1	0
Property Equiv.	1:1	7	5	5
Property Typecasting Subsum.	1:n	5	3	0
Property Chain Equiv.	m:n	26	15	13
Property Chain Subsum.	m:n	17	7	0

step to load large-size ontology pairs. The generated alignments in EDOAL format are available at this link.[4]

2 Results

Since this is the first-year participation, AROA alignment system only evaluates its performance on the GeoLink benchmark. We will evaluate on other benchmarks in the near future. In the GeoLink benchmark, there are 19 simple mappings, including 10 class equivalences, 2 class subsumption, and 7 property equivalences. And there are 48 complex mappings, including 5 property subsumption, 26 property chain equivalences, and 17 property chain subsumption. Table 6 shows alignment patterns and categories in the GeoLink Benchmark and the results of AROA system. We list the numbers of identified mappings for each pattern. There are two dimensions that we can look into the performance. One is the entity identification, which means, given an entity in the source ontology, the system should be able to generate related entities in the target ontology. Another dimension is relationship identification, which the system should detect the correct the relationship between these entities, such as equivalence and subsumption. Therefore, we list the number of correct entities and the number of correct relationships in order to help the reader to understand the strengths and weaknesses of the system. For example, In the Table 6, AROA correctly identifies all 1:1 class equivalence including entity and relationship. However, AROA also finds one class subsumption alignment, which is the class $PortCall$ in the GeoLink Base Ontology (GBO) is related to the class Fix in the GeoLink Modular Ontology (GMO). However, it outputs the relationship between $PortCall$ and Fix as equivalence, which it should be subsumption. Therefore, we count the number of correct entities as 1 and number of correct relations as 0. This criterion is also applied to other patterns. In addition, we compare the performance of AROA against other alignment systems in Table 7. And AROA achieved the best performance in terms of relaxed recall and f-measure.[5]

[4] http://oaei.ontologymatching.org/2019/results/complex/geolink/geolink_results.zip
[5] http://oaei.ontologymatching.org/2019/results/complex/geolink/index.html

Table 7. The Performance Comparison on GeoLink Benchmark

Matcher	AMLC	AROA	CANARD	LogMap	LogMapKG	LogMapLt	POMAP++
Relaxed_Precision	0.50	0.86	0.89	0.85	0.85	0.69	**0.90**
Relaxed_Recall	0.23	**0.46**	0.39	0.18	0.18	0.25	0.16
Relaxed_F-measure	0.32	**0.60**	0.54	0.29	0.29	0.36	0.26

3 General comments

From the performance comparison, only AROA and CANARD [5] can generate almost correct complex alignment, which means some alignments found by these two systems may not be completely correct, but it can be easily improved by semi-automated fashion. For example, the system can produce correct entities that should be involved in a complex alignment, but it doesn't output the correct relationship. Another situation is that the system can detect the correct relationship but fails to find all the entities. Based on these situations, we will investigate the incorrect alignments and improve the algorithm to find the relationship and entities as accurate as possible.

4 Conclusions

This paper introduces the AROA ontology alignment system and its preliminary results in the OAEI 2019 campaign. This year, AROA evaluates its performance on GeoLink benchmark and achieves the best performance in terms of recall and f-measure. We will continue to evaluate AROA on other benchmarks and improve the algorithm in the near future.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB'94, Proc. of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile. pp. 487–499 (1994), http://www.vldb.org/conf/1994/P487.PDF
2. Han, J., et al.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. **8**(1), 53–87 (2004). https://doi.org/10.1023/B:DAMI.0000005258.31418.83, https://doi.org/10.1023/B:DAMI.0000005258.31418.83
3. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press (1991)
4. Ritze, D., Meilicke, C., Sváb-Zamazal, O., Stuckenschmidt, H.: A pattern-based ontology matching approach for detecting complex correspondences. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009 (2009), http://ceur-ws.org/Vol-551/om2009_Tpaper3.pdf
5. Thiéblin, É., et al.: CANARD complex matching system: results of the 2018 OAEI evaluation campaign. In: Proc. of the 13th Int. Workshop on Ontology Matching, OM@ISWC 2018, Monterey, CA, USA, Oct. 8, 2018. pp. 138–143 (2018), http://ceur-ws.org/Vol-2288/oaei18_paper4.pdf

6. Zhou, L., et al.: A complex alignment benchmark: Geolink dataset. In: The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II. pp. 273–288 (2018). https://doi.org/10.1007/978-3-030-00668-6_17, https://doi.org/10.1007/978-3-030-00668-6_17

CANARD Complex Matching System: Results of the 2019 OAEI Evaluation Campaign*

Elodie Thiéblin, Ollivier Haemmerlé, Cassia Trojahn

IRIT & Université de Toulouse 2 Jean Jaurès, Toulouse, France
{firstname.lastname}@irit.fr

Abstract. This paper presents the results from the CANARD system in the OAEI 2019 campaign. CANARD is a system able to generate complex alignments. It is based on the notion of competency questions for alignment, as a way of expressing user needs. The system has participated in tracks where instances are available (populated Conference and Taxon datasets). This is the second participation of CANARD in the OAEI campaigns.

1 Presentation of the system

1.1 State, purpose, general statement

The CANARD (Complex Alignment Need and A-box based Relation Discovery) system discovers complex correspondences between populated ontologies based on Competency Questions for Alignment (CQAs). CQAs represent the knowledge needs of a user and define the scope of the alignment [4]. They are competency questions that need to be satisfied over two or more ontologies. Our approach takes as input a set of CQAs translated into SPARQL queries over the source ontology. The answer to each query is a set of instances retrieved from a knowledge base described by the source ontology. These instances are matched with those of a knowledge base described by the target ontology. The generation of the correspondence is performed by matching the subgraph from the source CQA to the lexically similar surroundings of the target instances.

In comparison with last year's version [3], CANARD can now deal with *binary* CQAs, *i.e.*, CQAs whose expected answers are pairs of instances or literal values. Last year it could only deal with *unary* CQAs (*i.e.*, CQAs whose expected answers are sets of instances). For example, here are examples of unary, binary and N-ary CQAs:

- A *unary* CQA expects a set of instances or values, *e.g.*, *Which are the accepted paper? (paper1), (paper2).*
- A *binary* CQA expects a set of instances or value pairs, *e.g.*, *Who wrote which paper? (person1, paper1), (person2, paper2).*

– An *n-ary* CQA expects a tuple of size 3 or more, *e.g.*, *What is the rate associated with which review of which paper? (paper1, review1, weak accept), (paper1, review2, reject).*

1.2 Specific techniques used

The approach has not changed much from last year [3]. The main difference with respect to binary CQAs is in Step (4), where two instances of the pair answer are matched instead of one (as in the case of unary CQAs), Step (5) and Step (8) which deal with the subgraph extraction and pruning.

The approach is detailed in the following steps over an example: the CQA expressed as a SPARQL query over the source knowledge base is:
`SELECT ?x ?y WHERE { ?x o1:paperWrittenBy ?y. }`

(1) Extract source DL formula e_s (*e.g.*, o_1*:paperWrittenBy*) from the SPARQL query.
(2) Extract lexical information from the CQA, L_s set labels of atoms from the DL formula (*e.g.*, "paper written by").
(3) Extract source answers ans_s of the CQA (*e.g.*, a pair of instances *(*o_1*:paper1,* o_1*:person1)*).
(4) Find equivalent or similar target answers ans_t to the source instances ans_s (*e.g.* o_1*:paper1* $\sim$ o_2*:paper1* and o_1*:person1* $\sim$ o_2*:person1*).
(5) Retrieve the subgraphs of target answers: for a binary query, it is the set of paths between two answer instances as well as the types of the instances appearing in the path (*e.g.*, a path of length 1 is found between o_2*:paper1* and o_2*:person1*). The path is composed of only one property and there are no other instances than o_2*:paper1* and o_2*:person1* in this path. Their respective types are retrieved: (o_2*:Paper*,o_2*:Document*) for o_2*:paper1* and (o_2*:Person*) for o_2*:person1*.
(6) For each subgraph, retrieve L_t the labels of its entities (*e.g.*, o_2*:writes* $\rightarrow$ "writes", o_2*:Person* $\rightarrow$ "person", o_2*:Paper* $\rightarrow$ "paper", *etc.*).
(7) Compare L_s and L_t.
(8) Select the subgraph parts with the best score, transform them into DL formulae. Keep the best path variable types if their similarity is higher than a threshold. (*e.g.*, the best type for the instance o_2*:paper1* is o_2*:Paper* because its similarity with the CQA labels is higher than the similarity of o_2*:Document*).
(9) Filter the DL formulae based on their confidence score (if their confidence score is higher than a threshold).
(10) Put the DL formulae e_s and e_t together to form a correspondence (*e.g.*, $\langle$ o_1*:paperWrittenBy* , *dom(*o_2*:Paper)* $\sqcap$ o_2*:writes*$^-$, $\equiv$ $\rangle$ and express this correspondence in a reusable format (*e.g.*, EDOAL). The confidence assigned to a correspondence is the similarity score of the DL formula computed.

The instance matching phase (Step (4)) is based on existing *owl:sameAs*, *skos:closeMatch*, *skos:exactMatch*. In case these links are not available, and exact label matching is applied instead.

Finding a subgraph (Step (5) and (8)) for a pair of instances consists in finding a path between the two instances. The shortest paths are considered more accurate. Because finding the shortest path between two entities is a complex problem, paths of length below a threshold are sought. First, paths of length 1 are sought, then if no path of length 1 is found, paths of length 2 are sought, *etc.* If more than one path of the same length are found, all of them go through the following process. When a path is found, the types of the instances forming the path are retrieved. If the similarity of the most similar type to the CQA is above a threshold, this type is kept in the final subgraph.

For example, for a "*paper written by*" CQA with the answer (o_2*:paper1*,o_2*:person1*) in the target knowledge, a subgraph containing the following triples is found:

1. $\langle$ o_2*:person1* , o_2*:writes* , o_2*:paper1* $\rangle$
2. $\langle$ o_2*:paper1* , *rdf:type* , o_2*:Paper* $\rangle$
3. $\langle$ o_2*:paper1* , *rdf:type* , o_2*:Document* $\rangle$
4. $\langle$ o_2*:person1* , *rdf:type* , o_2*:Person* $\rangle$

The most similar type of o_2*:person1* is o_2*:Person*, which is below the similarity threshold. Triple 4 is then removed from the subgraph. The most similar type of o_2*:paper1* is o_2*:Paper*. Triple 3 is therefore removed from the subgraph. o_2*:Paper*'s similarity is above the similarity threshold: triple 2 stays in the subgraph. The translation of a subgraph into a SPARQL query is the same for binary and unary CQAs. Therefore, the subgraph will be transformed into a SPARQL query and saved as the following DL formula: *dom*(o_2*:Paper*) $\sqcap$ o_2*:writes*$^-$.

The similarity between the sets of labels L_s and L_t of Step (7) is the cartesian product of the string similarities between the labels of L_s and L_t (equation 1).

$$sim(L_s, L_t) = \sum_{l_s \in L_s} \sum_{l_t \in L_t} strSim(l_s, l_t) \tag{1}$$

$strSim$ is the string similarity between two labels l_s and l_t (equation 2). τ is the threshold for the similarity measure. In our experiments, we have empirically set up $\tau = 0.5$. $\tau = 0.5$ in our implementation.

$$strSim(l_s, l_t) = \begin{cases} \sigma \text{ if } \sigma > \tau, \text{ where } \sigma = 1 - \dfrac{levenshteinDist(l_s, l_t)}{\max(|l_s|, |l_t|)} \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

The confidence value score of a correspondence (Step (9)) is calculated with the following equation, then truncated to 1:

$$confidence = labelSim + structuralSim \tag{3}$$

Label similarity *labelSim* is the sum of the label similarity of each entity of the formula with the CQA.

Structural similarity *structSim*. This similarity was introduced to enhance some structural aspects in a formula. In the implementation of the approach, this value is set to 0.5 when a path between the two instances of the answer, and 0 for a unary CQA subgraph. Indeed, if the label similarity of the path is 0, the structural similarity hints that the fact that a path was found is a clue in favour of the resulting DL formula.

1.3 Adaptations made for the evaluation

Automatic generation of CQAs OAEI tracks do not cover CQAs i.e., the CQAs can not be given as input in the evaluation. We extended last year's query generator so that it can output binary queries. The query generator now produces three types of SPARQL queries: *Classes*, *Properties* and *Property-Value pairs*.

Classes For each *owl:Class* populated with at least one instance, a SPARQL query is created to retrieve all the instances of this class. If `<o1#class1>` is a populated class of the source ontology, the following query is created:
`SELECT DISTINCT ?x WHERE {?x a <o1#class1>.}`

Properties For each *owl:ObjectProperty* or *owl:Dataproperty* with at least one instantiation in the source knowledge base, a SPARQL query is created to retrieve all instantiations of this property. If `<o1#property1>` is an instantiated property of the source ontology, the following query is created:
`SELECT DISTINCT ?x ?y WHERE {?x <o1#property1> ?y.}`

Property-Value pairs Inspired by the approaches of [1,2,5], we create SPARQL queries of the form

- `SELECT DISTINCT ?x WHERE {?x <o1#property1> <o1#Value1>.}`
- `SELECT DISTINCT ?x WHERE {<o1#Value1> <o1#property1> ?x.}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value".}`

These property-value pairs are computed as follow: for each property (object or data property), the number of distinct object and subject values are retrieved. If the ratio of these two numbers is over a threshold (arbitrarily set to 30) and the smallest number is smaller than a threshold (arbitrarily set to 20), a query is created for each of the less than 20 values. For example, if the property `<o1#property1>` has 300 different subject values and 3 different object values (`"Value1"`, `"Value2"`, `"Value3"`), the ratio $|subject|/|object| = 300/3 > 30$ and $|object| = 3 < 20$. The 3 following queries are created as CQAs:

- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value1".}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value2".}`
- `SELECT DISTINCT ?x WHERE {?x <o1#property1> "Value3".}`

The threshold on the smallest number ensures that the property-value pairs represent a category. The threshold on the ratio ensures that properties represent categories and not properties with few instantiations.

Implementation adaptations In the initial version of the system, Fuseki server endpoints are given as input. For the SEALS evaluation, we embedded a Fuseki server inside the matcher. The ontologies are downloaded from the SEALS repository, then uploaded in the embedded Fuseki server before the matching process can start. This downloading-uploading phase takes time, in particular when dealing with large files.

The CANARD system in the SEALS package is available at `http://doi.org/10.6084/m9.figshare.7159760.v2`. The generated alignments in EDOAL format are available at:

- **Populated Conference**: `http://oaei.ontologymatching.org/2019/results/complex/popconf/populated_conference_results.zip`
- **GeoLink**: `http://oaei.ontologymatching.org/2019/results/complex/geolink/geolink_results.zip`
- **Taxon**: `http://oaei.ontologymatching.org/2019/results/complex/taxon/results_taxon_2019.zip`

In this year's OAEI complex track, the Populated Conference, GeoLink and Taxon subtracks provide datasets with common instances. CANARD could generate alignments on these three datasets.

2 Results

2.1 Populated Conference

CANARD achieves this task with the longest runtime (96 min). The number of correspondences output by CANARD is detailed in Table 1. The results are detailed in Table 2.

CANARD achieves the highest the best query Fmeasure CQA Coverage score. AMLC achieves the best classical CQA Coverage, CANARD the second best. Both achieve CQA Coverage scores above ra1, but CANARD does not rely on an input alignment (in opposite to AMLC).

The classical Precision of CANARD is the lowest, its query Fmeasure precision above that of AMLC.

2.2 GeoLink

The number of correspondences output by CANARD is detailed in Table 3. The results are detailed in Table 4.

Relaxed precision and recall scores are calculated based on how the entities in the output correspondences are similar to those in the reference correspondences. All multiplied by a coefficient given the relation of the output correspondence and that of the reference one.

CANARD achieves the second best relaxed precision score, behind POMAP++ and the second best relaxed recall score behind AROA.

Table 1: Number of correspondences output by CANARD over the Populated Conference dataset

pair	(1:1)	(1:n)	(m:1)	(m:n)	Total
cmt-conference	19	100	0	5	124
cmt-confOf	18	17	0	6	41
cmt-edas	22	59	2	12	95
cmt-ekaw	11	111	0	12	134
conference-cmt	17	80	0	7	104
conference-confOf	28	13	3	0	44
conference-edas	17	38	0	8	63
conference-ekaw	31	120	2	3	156
confOf-cmt	15	37	0	0	52
confOf-conference	14	22	0	0	36
confOf-edas	15	36	0	0	51
confOf-ekaw	14	39	0	0	53
edas-cmt	20	50	0	4	74
edas-conference	16	49	0	2	67
edas-confOf	24	28	1	0	53
edas-ekaw	18	121	0	4	143
ekaw-cmt	15	71	0	0	86
ekaw-conference	31	80	0	0	111
ekaw-confOf	13	16	0	0	29
ekaw-edas	30	55	0	1	86
TOTAL	388	1142	8	64	1602

2.3 Taxon

CANARD has the longest runtime over the Taxon dataset (512 minutes $\sim$ 8h32). It is longer than last year's rutime (42 minutes) because the inclusion of binary queries in the process increases the number of input queries. Moreover the path finding algorithm consists in looking for all possible paths between two instances relies on SPARQL queries which take a long time to be executed.

The number of correspondences output by CANARD is detailed in Table 1. The results are detailed in Table 2.

Last year, CANARD had output 142 correspondences. This year it has output 791.

CANARD achieves the best CQA Coverage scores over the Taxon dataset. This year, the evaluation was oriented. For example, let's take a set of equivalent correspondences: Q=⟨SELECT ?x WHERE{ ?x a agtx:Taxon}, SELECT ?x WHERE{ ?x a dbo:Species}⟩. If an output alignment *agronomicTaxon-dbpedia* contains ⟨ *agtx:Taxon , dbo:Species ,* ≡ ⟩ but the alignment *dbpedia-agronomicTaxon* does NOT contain ⟨ *dbo:Species , agtx:Taxon ,* ≡ ⟩. The coverage score of Q for the pair *agronomicTaxon-dbpedia* is 1 but the coverage score of Q for *dbpedia-agronomicTaxon* is 0. Last year the evaluation was non-oriented, so the coverage score of Q would be the same (1.0) for both pairs. Taking that into consideration, we computed that if the evaluation was oriented this year, the classical

Table 2: Results of CANARD over the Populated Conference dataset

	Coverage		Precision		
pair	classical	query Fmeasure	classical	query Fmeasure	not disjoint
cmt-conference	0.28	0.53	0.15	0.48	0.90
cmt-confOf	0.50	0.50	0.22	0.60	0.98
cmt-edas	0.65	0.65	0.14	0.42	0.97
cmt-ekaw	0.35	0.59	0.07	0.39	0.97
conference-cmt	0.41	0.45	0.14	0.50	0.85
conference-confOf	0.30	0.35	0.25	0.55	0.73
conference-edas	0.36	0.38	0.19	0.41	0.79
conference-ekaw	0.38	0.47	0.19	0.45	0.78
confOf-cmt	0.50	0.71	0.19	0.76	1.00
confOf-conference	0.27	0.40	0.39	0.73	1.00
confOf-edas	0.23	0.28	0.14	0.45	0.67
confOf-ekaw	0.29	0.42	0.17	0.43	0.83
edas-cmt	0.59	0.67	0.27	0.54	0.97
edas-conference	0.39	0.53	0.37	0.62	0.97
edas-confOf	0.33	0.39	0.21	0.39	0.60
edas-ekaw	0.62	0.72	0.16	0.45	0.87
ekaw-cmt	0.43	0.58	0.30	0.58	0.92
ekaw-conference	0.30	0.50	0.30	0.62	0.93
ekaw-confOf	0.23	0.33	0.31	0.61	0.93
ekaw-edas	0.58	0.64	0.10	0.46	0.92
Average	0.40	0.51	0.21	0.52	0.88

Table 3: Number of correspondences output by CANARD over the GeoLink dataset

pair	(1:1)	(1:n)	(m:1)	(m:n)	Total
gbo-gmo	14	17	13	1	45
gmo-gbo	12	3	0	0	15

CQA Coverage of CANARD would have been 0.197, which shows significant improvement over last year's result: 0.13.

Some correspondences such as $\langle$ *agronomicTaxon:FamilyRank* , $\exists$ *dbo:family*$^{-}$*.wikidata:Q756* , $\equiv \rangle$ or $\langle$ *agronomicTaxon:GenusRank* , $\exists$ *dbo:genus*$^{-}$*.wikidata:Q756* , *(* $\rangle$wikidata:Q756 being the Plant class in wikidata) have more specific target members because of the Plant type restriction. Such correspondences entail higher precision-oriented CQA Coverage and Precision scores than classical ones.

3 General comments

CANARD relies on common instances between the ontologies to be aligned. Hence, when such instances are not available, as for the Hydrography datasets, the approach is not able to generated complex correspondences. Furthermore, CANARD is need-oriented and requires a set competency questions to guide the

Table 4: Results of CANARD over the Populated Conference dataset

Relaxed Precision	Relaxed Fmeasure	Relaxed Recall
0.85	0.59	0.46

Table 5: Number of correspondences output by CANARD over the Taxon dataset

pair	(1:1)	(1:n)	(m:1)	(m:n)	Total
agronomicTaxon-agrovoc	3	25	0	0	28
agronomicTaxon-dbpedia	10	38	0	0	48
agronomicTaxon-taxref	4	28	0	0	32
agrovoc-agronomicTaxon	0	6	4	23	33
agrovoc-dbpedia	3	33	2	21	59
agrovoc-taxref	0	0	0	0	0
dbpedia-agronomicTaxon	5	62	4	26	97
dbpedia-agrovoc	8	57	0	29	94
dbpedia-taxref	18	198	0	29	245
taxref-agronomicTaxon	9	26	0	13	48
taxref-agrovoc	2	17	0	5	24
taxref-dbpedia	5	50	5	23	83
TOTAL	67	540	15	169	791

matching process. Here, these "questions" have been automatically generated based on a set of patterns.

In comparison to last year's campaign, CANARD can now deal with binary CQAs in the form of SPARQL queries with two variables in the SELECT clause.

CANARD's runtime is extremely long. It depends (among other things) on the performance of the SPARQL endpoint it interrogates and the presence (or not) of equivalent links.

However, even with generated queries (instead of user input CQAs) it obtains some of the best coverage scores.

4 Conclusions

This paper presented the adapted version of the CANARD system and its preliminary results in the OAEI 2019 campaign. This year, we have been participated in the Taxon, Populated Conference and GeoLink track, in which ontologies are populated with common instances. CANARD was the only system to output complex correspondences on the Taxon track.

Acknowledgements

Elodie Thiéblin has been funded by Pôle Emploi for the redaction of this paper. The authors have also been partially supported by the CNRS Blanc project RegleX-LD.

Table 6: Results of CANARD over the Taxon dataset

	CQA Coverage				Precision			
pair	classical	rec.-or.	prec.-or.	overlap	classical	re.-or.	prec.-or.	overlap
agronomicTaxon-agrovoc	0	0.67	0.33	0.83	0.14	0.64	0.39	1.00
agronomicTaxon-dbpedia	0	0.42	0.58	0.83	0.06	0.40	0.42	0.98
agronomicTaxon-taxref	0.33	0.50	0.42	0.50	0.28	0.76	0.57	1.00
agrovoc-agronomicTaxon	0.17	0.17	0.17	0.17	0.12	0.79	0.50	0.91
agrovoc-dbpedia	0.17	0.17	0.17	0.17	0.07	0.27	0.22	0.58
agrovoc-taxref	0	0	0	0	NaN	NaN	NaN	NaN
dbpedia-agronomicTaxon	0.17	0.17	0.17	0.17	0.06	0.53	0.56	0.89
dbpedia-agrovoc	0.17	0.17	0.17	0.17	0.03	0.47	0.36	0.78
dbpedia-taxref	0.17	0.17	0.17	0.17	0.03	0.21	0.16	0.94
taxref-agronomicTaxon	0.33	0.50	0.42	0.50	0.04	0.31	0.24	1.00
taxref-agrovoc	0.17	0.42	0.42	0.50	0.04	0.33	0.28	1.00
taxref-dbpedia	0	0.08	0.17	0.33	0.04	0.30	0.30	0.99
Average	0.14	0.28	0.26	0.36	0.08	0.45	0.36	0.91

References

1. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: ISWC. pp. 598–614. Springer (2010)
2. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Discovering concept coverings in ontologies of linked data sources. In: ISWC. pp. 427–443. Springer (2012)
3. Thiéblin, É., Haemmerlé, O., Trojahn, C.: CANARD complex matching system: results of the 2018 OAEI evaluation campaign. In: Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 138–143 (2018), http://ceur-ws.org/Vol-2288/oaei18_paper4.pdf
4. Thiéblin, E., Haemmerlé, O., Trojahn, C.: Complex matching based on competency questions for alignment: a first sketch. In: Ontology Matching Workshop. p. 5 (2018)
5. Walshe, B., Brennan, R., O'Sullivan, D.: Bayes-recce: A bayesian model for detecting restriction class correspondences in linked open data knowledge bases. International Journal on Semantic Web and Information Systems 12(2), 25–52 (2016)

DOME Results for OAEI 2019

Sven Hertling[0000−0003−0333−5888] and Heiko Paulheim[0000−0003−4386−8195]

Data and Web Science Group, University of Mannheim, Germany
{sven,heiko}@informatik.uni-mannheim.de

Abstract. DOME (Deep Ontology MatchEr) is a scalable matcher for instance and schema matching which relies on large texts describing the ontological concepts. The doc2vec approach is used to generate a vector representation of the concepts based on the textual information contained in literals. The cosine distance between two concepts in the embedding space is used as a confidence value. In comparison to the previous version of DOME it uses an instance based class matching approach. Due to its high scalability, it can also produce results in the largebio track of OAEI and can be applied to very large knowledge graphs. The results look promising if huge texts are available, but there is still a lot of room for improvement.

Keywords: Ontology Matching · Knowledge Graph · Doc2Vec

1 Presentation of the system

Ontology matching is a key feature for the semantic web vision because it allows to use and interpret datasets which are unknown at the time of writing knowledge accessing software. [11] shows that there are many different elementary matching approaches on element, structure and instance levels. The `Deep Ontology MatchEr` (DOME) focuses at element and instance level matching. One of the reasons is that there are more and more instance matching tracks at the OAEI (Ontology Alignment Evaluation Initiative) like `SPIMBENCH`, `Link Discovery`, and `Knowledge graph`. These tracks need a scalable matching system. Thus, the main signal for finding correspondences is string based. Many other knowledge graphs in the Linked Open Data Cloud [2] also have a lot of literals with long texts which can be optimally used by the matching framework presented in this paper. Especially knowledge graphs extracted from Wikipedia such as DBpedia [1] or YAGO [5] contains descriptions of resources (abstracts of wiki pages).

1.1 State, purpose, general statement

The overall matching strategy of DOME is shown in figure 1. It starts with a simple string matching followed by a confidence adjustment. This is applied for

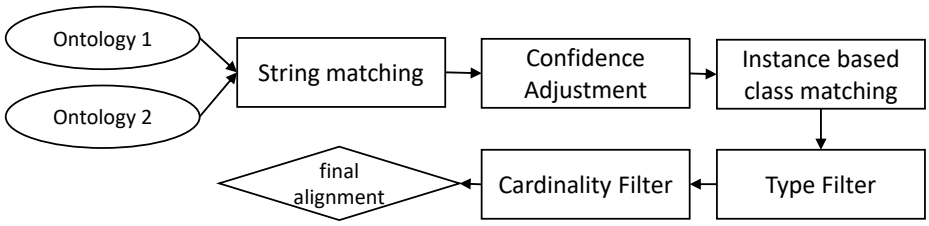


Fig. 1. Overview of the DOME matching strategy.

all classes, instances, and properties. The latter one includes *owl:ObjectProperty*, *owl:DatatypeProperty*, and *rdf:Property* (as retrived by the jena[1] method *OntModel.listAllOntProperties()*). As a next step in the pipeline, an instance based class matching is applied. It uses all matched individuals and based on those types, tries to find meaningful class mappings.

The following type filter deletes all correspondences where the type of source and target concept is different (like *owl:DatatypeProperty - owl:ObjectProperty*). This might happen because all properties (also *rdf:Property*) can be matched with each other. The final cardinality filter ensures a one to one mapping by sorting the correspondences by confidence and iterates over them in descending order. If the source or target entity is not already matched, it counts a valid correspondence - otherwise it will be dropped and will not appear in the final alignment.

In the following, the first three matching stages of DOME are discussed in more detail.

String matching As shown in figure 2, DOME uses multiple properties for matching all types of resources. If a *rdfs:label* from ontology A matches the *rdfs:label* from a resources in ontology B after the preprocessing, DOME creates a mapping with a static confidence of 1.0. The same confidence is applied when a *skos:prefLabel* matches. In case a URI fragment or *skos:altLabel* fits, a lower confidence of 0.9 is used.

The string preprocessing consists of tokenizing the text (also takes care of CamelCase[2] formatting), stopword removal and lowercasing. Afterwards the text is concatenated together to form a new textual representation. In case the initial text contains mostly numbers, the whole text is discarded.

Confidence Adjustment The confidence adjustment stage of DOME iterates over all correspondences and reassign a new confidence in case it is possible. The main approach used here is doc2vec [7] which is based on word2vec [8]. It allows to compare texts of different lengths and represent them as a fixed length vector. A comparison of these vectors can be achieved with a cosine similarity.

[1] `https://jena.apache.org`
[2] `https://en.wikipedia.org/wiki/Camel_case`

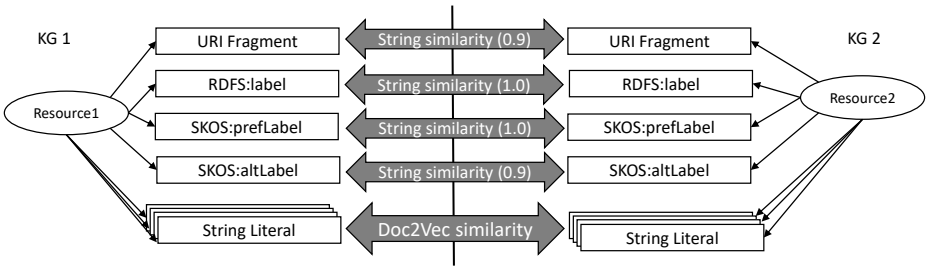


Fig. 2. DOME literal comparisons.

In comparison to DOME submitted to OAEI 2018, the generation of the text for a given resource has changed. In the current version, all statements in the ontology are examined where a given resource has the subject position. If the object is a literal and the datatype of it corresponds to *xsd:string* or *rdf:langString* or contains a language tag, it will be selected. All those literals are preprocessed in the same way as described in paragraph *string matching* and concatenated together. This text forms a document which is used for training a doc2vec model. DOME uses the DM sequence learning algorithm with a vector size of 300 and window size of 5 as in the previous version of this matcher dla[3]. The minimal word frequency is set to one to allow all words contribute to the concept vector. The adjusted confidence is later used in the cardinality filter to create a 1:1 mapping.

Instance based class matching After the class, instance, and property matching an additional class alignment step is performed. The basic idea is to inspect the types (classes) of already matched instances. If two individuals are the same, there is a high probability that some of the corresponding types should be also matched.

We experimented with three different similarity metrics for two given classes c_1 and c_2. The dice similarity metric [9] is defined as follows:

$$Sim_{DICE}(c_1, c_2) = \frac{2 * |I_{c_1} \cap I_{c_2}|}{|I_{c_1}| + |I_{c_2}|} \in [0...1]$$

I_{c_1} and I_{c_2} denotes the set of instances which have c_1 (c_2) as one of its type. $I_{c_1} \cap I_{c_2}$ corresponds to the matched instances which are typed with both c_1 and c_2. Sim_{DICE} corresponds to the overlap of matched instances with both classes and all instances of the two classes separately.

[6] also includes a more relaxed version of the previous similarity called Sim_{MIN} which is defined as

$$Sim_{MIN}(c_1, c_2) = \frac{|I_{c_1} \cap I_{c_2}|}{min(|I_{c_1}|, |I_{c_2}|)} \in [0...1]$$

It interrelates the matched instances with both classes and the instances of the smaller-sized class. As stated in [6] Sim_{DICE} is always smaller or equal to Sim_{MIN}.

A third possibility is Sim_{BASE} [6] which matches the classes c_1 and c_2 in case at least one instance with those classes is matched:

$$Sim_{BASE}(c_1, c_2) = \begin{cases} 1 & \text{if } |I_{c_1} \cap I_{c_2}| > 0 \\ 0 & \text{if } |I_{c_1} \cap I_{c_2}| = 0 \end{cases} \in [0...1]$$

After experimenting with those measures, it turned out that Sim_{BASE} introduces a lot of wrong correspondences because each error in the instance matching is directly forwarded to the class matches. Sim_{MIN} needed a very low threshold and ranks the classes suboptimal. Thus some similarity between Sim_{BASE} and Sim_{MIN} is needed. One possible way is to incorporate the quality of the matcher at hand - especially how many instance correspondences it finds. Thus another similarity called Sim_{MATCH} is used in DOME and defined as follows:

$$Sim_{MATCH}(c_1, c_2) = \frac{|I_{c_1} \cap I_{c_2}|}{|C_I|} \in [0...1]$$

where C_I represents all instance correspondences created by the matcher. The threshold is set to 0.01 meaning that 1 % of the matches should have the same class. If this is the case, the classes will be matched with a confidence of $Sim_{MATCH}(c_1, c_2)$. This value is rather low. All correspondences generated by this step are therefore scaled to minimum of 0.1 and maximum of 1.0.

1.2 Specific techniques used

The two main techniques used in DOME are the doc2vec approach [7] for comparing the textual representation of the resources and the instance based class matching component.

1.3 Adaptations made for the evaluation

As in the previous version of DOME for OAEI 2018 the DL4J[3] (Deep Learning for Java) library is used as an implementation of the doc2vec approach. Running DOME with this dependency is not easy in SEALS. Therefore we use MELT[4] to package our matcher. The framework generates an intermediate matcher which executes an external process (which is again in Java). This process runs now in its own Java virtual machine (JVM) and allows to load system dependent library files (files with *dll* or *so* extension). [3] explains in more detail why this is necessary.

1.4 Link to the system and parameters file

DOME can be downloaded from
`https://www.dropbox.com/s/1bpektuvcsbk5ph/DOME.zip?dl=0`.

[3] `https://deeplearning4j.org`

2 Results

This section discusses the results of DOME for each track of OAEI 2019 where the matcher is able to produce results. The following tracks are included: anatomy, conference, largebio, phenotype, and knowledge graph track.

Similar to the previous version of DOME, the current matcher is not able to match multiple languages and thus fail on multifarm track. Specific interfaces and matching strategies for the complex and interactive track are currently not implemented.

2.1 Anatomy

For the anatomy track, DOME uses the string comparison method which results in similar precision and recall as the baseline. Properties like *oboInOwl:hasRelatedSynonym* or *oboInOwl:hasDefinition* are used to generate a textual representation of the concepts but this does not introduce better confidence values.

DOME returns 948 correspondences. 932 matches with a confidence of 1.0 which are all correct. 12 correspondences scored with 0.9 are all false positives. Therefore a confidence filter would make sense for this specific track.

The presented matcher has a very low runtime and scales to very huge ontologies. The runtime of 23 seconds is the second best value in this track.

Due to a slightly lower recall (0.007) and precision (0.001) DOME has a lower F-Measure (0.006) than the baseline. The reason could be the different string preprocessing techniques.

2.2 Conference

In the following analysis we refer to the `rar2`[4] reference alignment because it contains more correspondences which are carefully resolved by an evaluator.

When matching classes DOME is same as the edna baseline. Most correspondences have a confidence of 0.9 because the conference track has mostly all textual information in URL fragments. Only one mapping is scored with 1.0 which is <*edas:Country*, *iasted:Conference_state*, =, 1.0>. It is generated by the instance based class matching because both contain *Mexico* as an individual. This mapping is a false positive. The instance based class matching could not help here, because in most of the test cases no instances are available. Properties are matched with an F1-measure of 0.22 which is better than the edna baseline but lower than 5 other matchers. In comparison to the old version of this matcher, the F1-measure is increased by 0.01. Figure 3 shows the result of DOME divided into test cases. It shows that in four test cases (where the source ontology is *confOf*) the matcher is not able to return true positive correspondences.

[4] `http://oaei.ontologymatching.org/2019/results/conference/index.html`

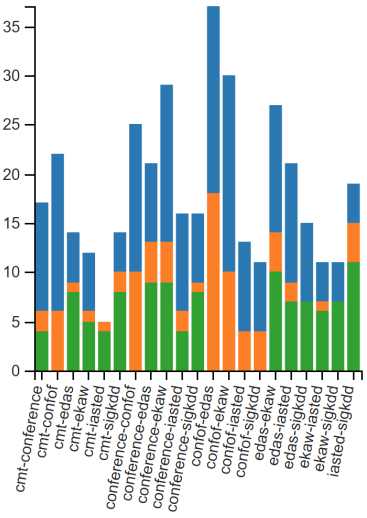


Fig. 3. Analysis of results for conference track. The x axis represents the test cases and y axis the amount of correspondences. Green bars indicates true positives, orange bars false positives, and blue bars false negatives. The plot is generated by MELT framework [4].

2.3 Largebio

As the name already suggests, the largebio track needs matchers which scale well. Test case four is a large test case which matches the whole FMA ontology with a large fragment of SNOMED. The source ontology has 78,989 classes and the target ontology 122,464 classes. This would result in more than 9 billion comparisons when doing it naively. The runtime of DOME for this test case is 38 seconds which is the second best runtime. Moreover DOME is able to complete all tasks within the given timeout.

In task 3, 4, 5, and 6 DOME has the highest precision of all matchers but misses a lot of correspondences in the gold standard and has therefore a lower recall. In task one and two matcher Wiktionary have a higher precision. F-measure wise DOME usually beats Wiktionary and AGM but AML and LogMap variants are better.

2.4 Phenotype

In phenotype track, the matcher should find alignments between disease and phenotype ontologies. The matcher has the highest precision of 0.997 together with FCAMapKG for test case HP-MP and second best for task DOID-ORDO. With the low recall of 0.303 and 0.426 the F-measure is around 0.465 and 0.596.

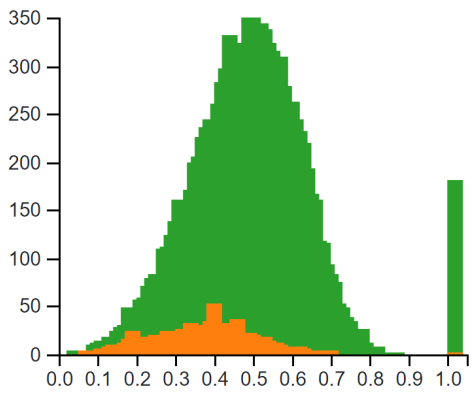


Fig. 4. Analysis of confidences in knowledge graph track. The x axis represents the confidence value and the y axis shows the amount of correspondences. Green bars indicates true positives and orange bars indicates false positives. False negatives are left out because they don't have any confidence assigned by the matching system. The plot is generated by MELT framework [4].

2.5 Knowledge Graph

In the second version of the knowledge graph track, the systems should be able to match classes, properties and instances. DOME was able to run 4 out of 5 test cases. The remaining test case could not be finished because of memory issues.

In comparison to the previous version of the track, classes are more difficult to match. DOME could achive an F-measure of 0.77 for classes (not counting the unfinished test case) and 0.96 for properties. Only FCAMap-KG and Wiktionary are better in matching the latter one. Instances are matched with a F-measure of 0.88 (again not counting the unfinished test case). In average DOME returns 22 class, 75 property, and 4,895 instance mappings.

3 General comments

3.1 Comments on the results

The discussion of the results shows that DOME is in a development phase. Some improvements are already incorporated and some further ideas are discussed in the next section.

3.2 Discussions on the way to improve the proposed system

One further improvment is still the ability to match different languages. As stated in [3] we could use cross lingual embeddings as shown in [10]. Another possibility would be to use a translation step in between.

The confidence adjustment step can not only be done with doc2vec based models but also with tf-idf or other document comparison methods. This should be tried out in future version of this matcher.

The memory issue in the knowledge graph track can be solved by writing all text representations of all resources on disk and train the doc2vec model on this file.

4 Conclusions

In this paper, we have analyzed the results of DOME in OAEI 2019. It shows that DOME is a highly scalable matcher which generates class, property and instance alignments. With the new component DOME is able to match classes based on instances and thus increase the recall of class alignments.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)
2. Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked data on the web (ldow2008). In: Proceedings of the 17th international conference on World Wide Web. pp. 1265–1266. ACM (2008)
3. Hertling, S., Paulheim, H.: Dome results for oaei 2018. In: OM@ ISWC. pp. 144–151 (2018)
4. Hertling, S., Portisch, J., Paulheim, H.: Melt - matching evaluation toolkit. In: SEMANTICS. Karlsruhe. (2019)
5. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence **194**, 28–61 (2013)
6. Kirsten, T., Thor, A., Rahm, E.: Instance-based matching of large life science ontologies. In: International Conference on Data Integration in the Life Sciences. pp. 172–187. Springer (2007)
7. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188–1196 (2014)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
9. van Rijsbergen, C.: Information retrieval (1979)
10. Ruder, S., Vulić, I., Søgaard, A.: A survey of cross-lingual word embedding models. arXiv preprint arXiv:1706.04902 (2017)
11. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV, Lecture Notes in Computer Science, vol. 3730, pp. 146–171. Springer Berlin Heidelberg (2005)

EVOCROS: Results for OAEI 2019

Juliana Medeiros Destro[1], Javier A. Vargas[1], Julio Cesar dos Reis[1], and Ricardo da S. Torres[2]

[1]Institute of Computing, University of Campinas, Campinas-SP, Brazil
[2]Norwegian University of Science and Technology (NTNU), Ålesund, Norway
{juliana.destro,jreis}@ic.unicamp.br
ricardo.torres@ntnu.no, jalvarm.acm@gmail.com

Abstract. This paper describes the updates in EVOCROS, a cross-lingual ontology alignment system suited to create mappings between ontologies described in different natural language. Our tool combines syntactic and semantic similarity measures with information retrieval techniques. The semantic similarity is computed via NASARI vectors used together with *BabelNet*, which is a domain-neutral semantic network. In particular, we investigate the use of rank aggregation techniques in the cross-lingual ontology alignment task. The tool employs automatic translation to a pivot language to consider the similarity. EVOCROS was tested and obtained high quality alignment in the Multifarm dataset. We discuss the experimented configurations and the achieved results in OAEI 2019. This is our second participation in OAEI.

Keywords: cross-lingual matching · semantic matching · background knowledge · ranking aggregation

1 Presentation of the system

There is a growing number of ontologies described in different natural languages. The mappings among different ontologies are relevant for the integration of heterogeneous data sources to facilitate the exchange of information between systems. EVOCROS is our approach to automatic cross-lingual ontology matching. In our previous participation, in OAEI 2018, EVOCROS employed a weighted combination of similarity and semantic measures. The new version, submitted in OAEI 2019, combines syntactic and semantic similarity measures with information retrieval techniques. In this section, we describe the modifications to the system and the implemented techniques.

1.1 State, purpose, general statement

EVOCROS is a cross-lingual ontology alignment tool. The newest version of the tool leverages supervised methods of ranking aggregation techniques exploiting

labeled information (*i.e.*, training data) and ground-truth relevance to boost the effectiveness of a new ranker. Our goal is to leverage rank aggregation in cross-lingual mapping, by generating ranked lists based on distinct similarity measurements between the concepts of source and target ontologies.

1.2 Specific techniques used

The tool is developed in Python 3 and uses learning to rank techniques implemented in the well-known library *RankLib*. We model the mapping problem as an information retrieval query. Figure 1 depicts the workflow of the proposed technique. The inputs are source and target ontologies written in Web Ontology Language (OWL). These ontologies are converted to objects. The first step is the pre-processing of the source and target input ontologies, converting them into owlready2 objects. Each concept of the source ontology is compared to all concepts of the target ontology.

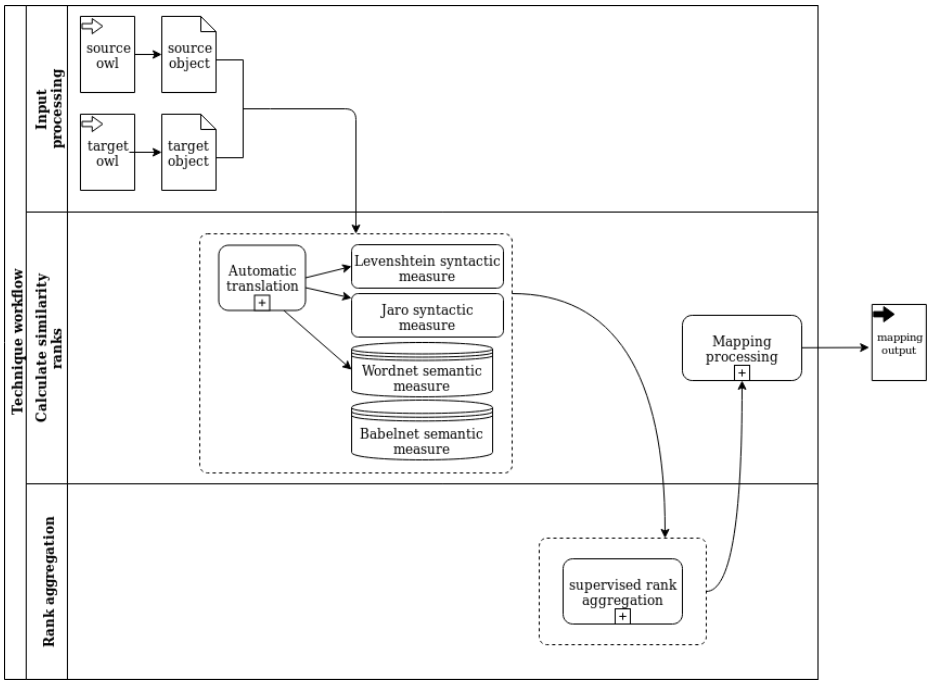


Fig. 1. General description of the technique. The mapping processing stage is where the top-1 entity of the final ranking is mapped to the input concept e_1.

RankLib: https://sourceforge.net/p/lemur/wiki/RankLib/ (As of November 16, 2019).

Python 3 library to manipulate ontologies as objects.

Each entity of the source ontology is compared with all entities of the same type found in the target ontology (*i.e.*, classes are matched to classes and properties are matched to properties). In this sense, for each entity e_i in the source ontology O_X, we calculate the similarity value with each entity e_j in the target ontology O_Y (Figure 2), thus generating a ranked list $\{rank1, rank2, rank3, rank4\}$ for each similarity measure used (*cf.* Figure 3).

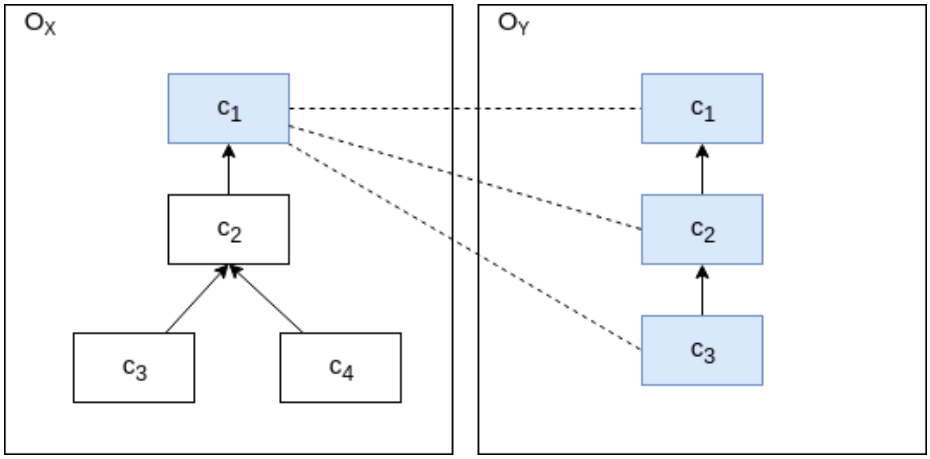


Fig. 2. Concept $c_1 \in O_X$ is compared against all concepts $c_n \in O_Y$.

For similarity measures that rely on monolingual comparison (*i.e.*, syntactic and WordNet), the automatic translation of labels of entities $e_i \in O_X$ and $e_j \in O_Y$ to a pivot language is used by leveraging Google Translate API during runtime. These similarity comparisons generate k ranks, each one based on a different similarity measure. We use the measures to generate the ranks, thus adding the flexibility to the use or the addition of different similarity measures without disrupting the technique.

The ranks are then aggregated using LambdaMART [7] because this technique has the best score among the majority of languages during the execution phase of OAEI 2019. Figure 4 presents that the set of multiple ranks are aggregated in a final rank. The Top-1 result of the aggregated rank $c_2 \in \mathcal{C}_{\mathcal{O}_\mathcal{Y}}$ is mapped to the source ontology entity $c_1 \in \mathcal{C}_{\mathcal{O}_\mathcal{X}}$, thus generating the candidate mapping $m(c_1, c_2)$ (*cf.* Figure 5). The mapping output follows the standard used by the Alignment API [**?**].

1.3 Link to the set of provided alignments (in align format)

Alignment results are available at https://github.com/jmdestro/evocros-results (As of November 16, 2019).

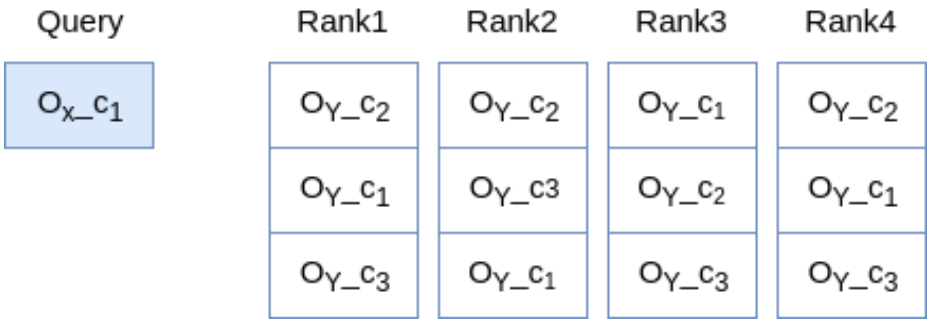


Fig. 3. Ranked lists generated by each similarity measure used.

2 Results

In this section, we describe the results obtained in the experiments conducted in OAEI 2019.

2.1 Multifarm

We consider the *MultiFarm* dataset [5], version released in 2015. Our experiments built cross-language ontology mappings by using English as a pivot language for Levenshtein [4], Jaro [3], and WordNet similarity measures. The semantic similarity relying on the *Babelnet* does not require a translation as it can retrieve the synsets used in NASARI vectors [1], by using the concepts original language. The application of each similarity measure in our technique generated a rank.

A subset of all languages was used for training and validation. The subsets are 10% of queries for training set, 15% queries for validation set, and 75% queries for testing. These subsets were generated per language and then combined, so the algorithms were trained, validated and tested using all languages at once. The comparable gold standard (*i.e.*, MultiFarm manually curated mappings) were adjusted to contain only the queries related to the testing subset. In this sense, a lower number of entities was considered in the tests, because we removed the set of queries used in training and validation from the reference mappings to ensure consistency.

Table 1 presents the obtained values for precision, recall, and f-measure for each language pair tested. The precision, recall, and f-measure scores have the same value due to the nature of the experiments. Our approach generates n : n mappings, where $n = |O_X| = |O_Y|$ because the ontologies are translations of each other to different natural languages, thus every entity in the source ontology presents a correspondence in the target ontology. In this sense, both the gold standard and the generated mappings have the same size because each query (*i.e.*, each entity in the source ontology) generates a mapping between the query (source entity) and the top-1 result of the final aggregated rank. Results

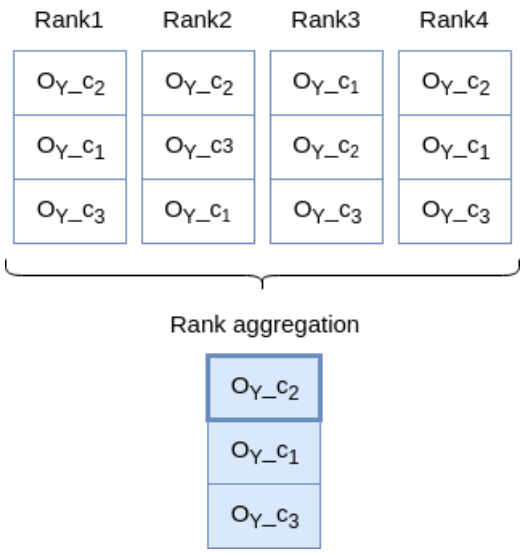


Fig. 4. Rank aggregation of the ranked lists. Each rank aggregation algorithm generates a distinct final rank.

Fig. 5. Mapping generated between source entity $c_1 \in O_X$ and top-1 entity of the final rank generated by the rank aggregation algorithm, $c_2 \in O_Y$.

show competitive results when compared to the other tools participating in the evaluation.

Table 1. Results achieved by different language pair

Language pair	Precision	Recall	F-measure
fr-nl	0.61290	0.61290	0.61290
en-pt	0.59140	0.59140	0.59140
es-nl	0.58065	0.58065	0.58065
cz-nl	0.52688	0.52688	0.52688
cn-pt	0.50538	0.50538	0.50538
es-ru	0.38710	0.38710	0.38710
cn-ru	0.32258	0.32258	0.32258
cz-ru	0.32258	0.32258	0.32258
de-ru	0.32258	0.32258	0.32258

3 General comments

In this section, we discuss our results and the ways to improve the system.

3.1 Comments on the results

The tool had satisfactory results, with competitive f-measure, but the execution time was exceedingly long due even with local caches for *Babelnet* NASARI vectors. This is due to the amount of comparisons required during execution because each concept or attribute in the source ontology is compared against all concepts and attributes of the target ontology.

3.2 Discussions on the way to improve the proposed system

This was the second evaluation of the system and results are encouraging. Our main goals for future work are: **Reduce execution time**: the tool has a long execution time even with local caches. Our future work will explore ontology partitioning during the pre-processing stage of the matching task to reduce the amount of comparisons needed, thus improving the execution time. **Bag of graphs**: ontologies can be represented as graphs, thus allowing for partitioning [2] and comparison of sub-graphs. Bag-of-graphs [6] is a graph matching approach, similar to bag-of-words. It represents graphs as feature vectors, highly simplifying the computation of graph similarity and reducing execution time. We propose as future investigation to use a simple vector-based representation for graphs and investigate it for cross-lingual ontology matching.

3.3 Comments on OAEI

Although we were not participating, our tool was executed on the Knowledge Graph track. There were issues during the evaluation phase, preventing the system to fully participate in both Multifarm and KG tracks.

4 Conclusion

The newest version of EVOCROS proposed an approach considering four similarity measures to build ranks and used a supervised method of rank aggregation. This is the second participation of the system in OAEI. The evaluation with the Multifarm dataset confirmed the quality of mappings generated by our technique. For future work, we plan to improve our cross-lingual alignment proposal by considering different combinations of similarity measures and different ways of computing the syntactic and semantic similarities taking into account additional stages in the pre-processing of the ontology.

Acknowledgements

This work was supported by São Paulo Research Foundation (FAPESP): grant #2017/02325-5.

References

1. Camacho-Collados, J., Pilehvar, M.T., Navigli, R.: Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. Artificial Intelligence **240**, 36–64 (2016)
2. Hamdi, F., Safar, B., Reynaud, C., Zargayouna, H.: Alignment-based partitioning of large-scale ontologies. In: Advances in knowledge discovery and management, pp. 251–269. Springer (2010)
3. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association **84**(406), 414–420 (1989)
4. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady **10**, 707–710 (1966)
5. Meilicke, C., GarcíA-Castro, R., Freitas, F., Van Hage, W.R., Montiel-Ponsoda, E., De Azevedo, R.R., Stuckenschmidt, H., ŠVáB-Zamazal, O., Svátek, V., Tamilin, A., et al.: Multifarm: A benchmark for multilingual ontology matching. Web Semantics: Science, Services and Agents on the World Wide Web **15**, 62–68 (2012)
6. Silva, F.B., de O. Werneck, R., Goldenstein, S., Tabbone, S., da S. Torres, R.: Graph-based bag-of-words for classification. Pattern Recognition **74**(Supplement C), 266 – 285 (Feb 2018). https://doi.org/10.1016/j.patcog.2017.09.018, http://www.sciencedirect.com/science/article/pii/S0031320317303680
7. Wu, Q., Burges, C.J., Svore, K.M., Gao, J.: Adapting boosting for information retrieval measures. Information Retrieval **13**(3), 254–270 (Jun 2010). https://doi.org/10.1007/s10791-009-9112-1

FCAMap-KG Results for OAEI 2019

Fei Chang[1], Guowei Chen[2,3], and Songmao Zhang[3]

[1] New York University, New York, USA
fc1271@nyu.edu
[2] University of Chinese Academy of Sciences
chenguowei17@mails.ucas.ac.cn
[3] Institute of Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, P.R. China
smzhang@math.ac.cn

Abstract. In OAEI 2016, we submitted the system FCA-Map for taking advantage of the Formal Concept Analysis (FCA) formalism in aligning large and complex biomedical ontologies. This year, we present a variant called FCAMap-KG, following the rationale of FCA-Map and designed for matching knowledge graphs. Among the 12 matchers participating in the OAEI 2019 Knowledge Graph track, our system ranks the first for instance and property mappings and ranks second for class mappings. As a result, FCAMap-KG has achieved the best overall F-measure for the track. This demonstrates the power of our FCA-based approach in identifying correspondences across different kinds of data and knowledge representation systems.

1 Presentation of the system

In OAEI 2016, we proposed the system FCA-Map [8,9,10] for taking advantage of the Formal Concept Analysis (FCA) formalism in aligning large and complex biomedical ontologies. Further in OAEI 2018, its variant FCAMapX [3] was submitted to largely improve the efficiency of the system. This year, we present a new variant called FCAMap-KG, for exploiting the potential of our FCA-based approach in matching knowledge graphs (KGs).

1.1 State, purpose, general statement

Formal Concept Analysis is a mathematical model for structuring concept hierarchies from clustering individuals [4,7]. In FCA, the domain or problem is described first by a formal context consisting of a set of objects, a set of attributes and their relations. Based on this, a lattice structure can be computed with each node representing a formal concept and edge a subconcept-superconcept relationship. Being a knowledge graph matching system based on FCA, FCAMap-KG follows the rationale of our previous systems FCA-Map and FCAMapX by consecutively constructing lexical and structural formal contexts and extracting mappings across KGs.

Ontologies and KGs are both knowledge representation models sharing RDF graphs as the underlying data structure. Ontologies focus on schematic knowledge and adopt logic-based reasoning to infer implied relations, whereas KGs mainly describe data in RDF triples and train numerical vector representations so as to predict semantic correlations [2]. Ontologies are limited to certain domains with precise knowledge and KGs are much larger in scale where data can be noisy. For both, identifying correspondences between systems is crucial for realizing semantic integration in the Semantic Web. Their distinctive differences, however, make applying ontology matching approaches to KGs a nontrivial endeavor. Particularly in our case, for building formal contexts at the structural level, FCA-Map and FCAMapX mainly use the ontological relationships including taxonomy, partonomy, disjointness, and property axioms among classes. While normally these are not available in KGs, FCAMap-KG turns to RDF triples where two instances are connected by a property. In our FCA-based approach, lexical formal context describes how names share tokens from which lexical mappings are generated. This is effective for both ontology and KG matching tasks as classes, properties and instances are all labeled with preferred names and synonyms.

1.2 Specific techniques used

The steps that FCAMap-KG system implements are presented as follows.

1. **Lexical matching.** For the given two KGs in comparison, the system builds three token-based formal contexts, for classes, properties and instances, respectively. One entity in KG can have multiple names and labels, and every one of them is treated as an object in the formal context; tokens extracted from all the names/labels in two KGs are used as attributes. Note that one object in the formal context can be associated with multiple entities in KGs and at the same time one entity can have multiple entries as objects. In the Galois lattice constructed from token-based formal context [4], lexical mappings are generated when formal concepts contain objects originated from two KGs.
2. **Structural matching.** The system proceeds to construct the structural formal context using lexical mappings obtained so far. KGs tend to have massive instances while properties and classes are much less, and as stated in [6,5], matching instances can be harder than classes and properties. Thus we focus on identifying structural correspondence among instances at this step. For the given two KGs, every instance is used as an object in the formal context. The attributes comes from pairing two RDF triples across KGS whose properties and tail instances have been matched, respectively, at the lexical step. Such a formal context describes how instances share connections to other instances, thus has a potential to reflect the structural similarities across KGs. In the lattice computed, structural mappings are generated when formal concepts contain instances from two KGs.
3. **Mapping filtering.** The OAEI 2019 Knowledge Graph track bases its evaluation on that all mappings are 1:1, i.e., one entity can only have at most one correspondence in the other KG. Due to this, the system employs a filtering process on cases

[4] We implemented the algorithm HERMES [1] for constructing the lattice.

when one entity occurs in multiple mappings identified. Mappings that possess more shared structural attributes and more lexical tokens are selected.

1.3 Adaptations made for the evaluation

Conforming to the evaluation criteria of the Knowledge Graph track this year, the SEALS submission of FCAMap-KG is modified to produce only 1:1 mappings. In general, FCAMap-KG is not restricted this way and can find cases when one entity is matched to multiple entities in another knowledge graph.

1.4 Link to the system and parameters file

The SEALS wrapped version of FCAMap-KG for OAEI 2019 is available at `https://drive.google.com/open?id=1pZ5Hzv8_wfULKYN4Uc_kcmlkPseJ7kQ_`

1.5 Link to the set of provided alignments

The results obtained by FCAMap-KG for OAEI 2019 are available at `https://drive.google.com/open?id=1bS19DDe7nZNC1MlHB8qX-yoACWBWELGR`

2 Results

In this section, we present the evaluation results obtained by running FCAMap-KG on *Knowledge Graph* track under the SEALS client in OAEI 2019 campaign. Although our system was not intended to participate in other tracks, OAEI reported whenever FCAMap-KG could generated an alignment [5]. Therefore, the results for these tracks will also be introduced including *Anatomy*, *Large Biomedical Ontologies*, *Disease and Phenotype*, and *Biodiversity and Ecology*. The evaluation was performed on a virtual machine (VM) with 32GB of RAM and 16 vCPUs (2.4 GHz).

2.1 The OAEI 2019 Knowledge Graph Track

The Knowledge Graph track requires finding alignments at both schema and data level, including class mappings, property mappings and instance mappings. The track consists of a total of five matching tasks among nine isolated knowledge graphs for describing movies, comics, TV and books. We follow the OAEI evaluation criteria in counting positives and negatives based on 1:1 matching and the partialness of gold standard. The overview results of FCAMap-KG are presented in Table 1 where **Size** indicates an average number of mappings obtained. As reported by OAEI [6], among the 12 participants, our system ranks the first in F-measure for instance and property mappings and ranks second for class mappings. As a result, FCAMap-KG has achieved the best overall F-measure for the track.

[5] http://oaei.ontologymatching.org/2019/results/

[6] http://oaei.ontologymatching.org/2019/results/knowledgegraph/index.html

Table 1. Overview results for Knowledge Graph track

Class				Property				Instance				Overall			
Size	Prec.	Rec.	F-m.	Size	Prec.	Rec.	F-m.	Size	Prec.	Rec.	F-m.	Size	Prec.	Rec.	F-m.
18.6	1.00	0.70	0.82	69.0	1.00	0.96	0.98	4530.6	0.90	0.79	0.84	4792.6	0.91	0.79	0.85

The overall performance of FCAMap-KG for each matching task is listed in Table 2, and when breaking down into class, property and instance mappings, the results for each task are shown by Table 3, Table 4, and Table 5, respectively. FCAMap-KG stands out in matching properties by having a 100% precision for four tasks, and according to OAEI, obtains the best F-measure for all five tasks among 12 participants. For instance mappings, the system stays in top three F-measures for all tasks; all the class mappings generated by FCAMap-KG for the track are correct and its F-measure ranks first for two tasks.

Table 2. Overall results for each matching task in Knowledge Graph track

Matching task	Size	Precision	Recall	F-measure
marvel cinematic universe - marvel	2,682	0.84	0.65	0.73
memory alpha - memory beta	13,171	0.92	0.85	0.88
memory alpha - stexpanded	3,174	0.94	0.89	0.91
star wars - swg	2,140	0.90	0.71	0.80
star wars - swtor	2,796	0.93	0.87	0.90

2.2 Other OAEI 2019 Tracks

OAEI reported the performance of our system in tracks other than the Knowledge Graph, and they are *Anatomy*, *Large Biomedical Ontologies*, *Disease and Phenotype*, and *Biodiversity and Ecology*. The results obtained by FCAMap-KG for these tracks are shown in Table 6, 7, 8, and 9, respectively.

- The Anatomy track aims at finding an alignment between the Adult Mouse Anatomy (2744 classes) and a fragment of the NCI Thesaurus (3304 classes) for describing the human anatomy.
- The Large Biomedical Ontologies track consists of identifying mappings among the Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). These ontologies are of both large-scale and semantic richness, and both whole ontologies and fragments are used.
- The Disease and Phenotype track involves the matching task between the Human Phenotype (HP) Ontology and the Mammalian Phenotype (MP) Ontology, and the

Table 3. Class results for each matching task in Knowledge Graph track

Matching task	Size	Precision	Recall	F-measure
marvel cinematic universe - marvel	8	1.00	1.00	1.00
memory alpha - memory beta	21	1.00	0.29	0.44
memory alpha - stexpanded	24	1.00	0.62	0.76
star wars - swg	12	1.00	0.80	0.89
star wars - swtor	28	1.00	0.80	0.89

Table 4. Property results for each matching task in Knowledge Graph track

Matching task	Size	Precision	Recall	F-measure
marvel cinematic universe - marvel	19	1.00	0.91	0.95
memory alpha - memory beta	93	1.00	0.94	0.97
memory alpha - stexpanded	73	0.98	0.98	0.98
star wars - swg	48	1.00	1.00	1.00
star wars - swtor	112	1.00	0.98	0.99

matching between Human Disease Ontology (DOID) and the Orphanet and Rare Diseases Ontology (ORDO).
- The Biodiversity and Ecology track aims at detecting equivalence between the Environment Ontology (ENVO) and the Semantic Web for Earth and Environment Technology Ontology (SWEET), and between the Plant Trait Ontology (PTO) and the Flora Phenotype Ontology (FLOPO).

Note that unlike FCAMap and FCAMapX specifically for aligning biomedical ontologies, FCAMap-KG targets knowledge graphs where schematic knowledge is generally rare, thus none domain thesauri or external terminologies have been used to facilitate the matching. It is understandable that FCAMap-KG did not perform as well as FCAMap and FCAMapX on life sciences ontologies. Nevertheless, without the support of any domain knowledge, FCAMap-KG ranks first in precision for MA-NCI task among 12 participants, for the two Disease and Phenotype tasks among 8 participants, and for ENVO-SWEET task among 6 participants.

3 General comments

3.1 Comments on the results

This is the third time that we participate in the OAEI campaign with our Formal Concept Analysis based system. Developed targeting knowledge graph matching, FCAMap-KG has achieved a satisfactory result by ranking first in F-measure for overall five KG tasks among 12 participants. For every single task, our system obtains the best F-measure

Table 5. Instance results for each matching task in Knowledge Graph track

Matching task	Size	Precision	Recall	F-measure
marvel cinematic universe - marvel	2,603	0.84	0.65	0.73
memory alpha - memory beta	12,474	0.92	0.85	0.88
memory alpha - stexpanded	3,008	0.94	0.89	0.91
star wars - swg	2,004	0.90	0.70	0.79
star wars - swtor	2,564	0.93	0.86	0.89

Table 6. Results for *Anatomy* track

Matching Task	Size	Precision	Recall	F-measure
MA-NCI	960	0.996	0.631	0.772

for property mappings and remains in top three for instance mappings. Of note, taking advantage of the efficiency mechanism implemented by FCAMapX, FCAMap-KG managed to finish all the KG tasks within given time despite the high computation complexity of FCA formalism per se. Additionally, although unintended, FCAMap-KG is reported in four biomedicine and ecology tracks by OAEI 2019 with a competitive performance in precision.

3.2 Discussions on possible improvements

The very first step of FCAMap-KG is lexical matching whose resultant mappings are used in the subsequent structural matching steps. This means that our system is susceptible to the lexical labeling of entities in knowledge graphs. When the naming is diverse across KGs, as in the case of *marvelcinematicuniverse - marvel*, gold standard mappings like $< marvelcinematicuniverse : Combat\_Enhancers,\ marvel : Adrenaline\_Pills >$ can be missed. For this task, FCAMap-KG's F-measure is 10% to 20% lower than the other four tasks, as listed in Table 2. This indicates the importance of structural matching which is capable of identifying matches not having anything common in names. We are in the process of constructing an iterative framework for using mappings obtained so far to enhance the current loop of matching until no further mappings are found. Such a comprehensive way of incorporating lexical and structural information of classes, properties and instances can take advantage of data and knowledge represented in KGs to the fullest.

As mentioned above, an adjustment made in FCAMap-KG for participating the Knowledge Graph track is to limit the mappings selected to one-to-one. Again, take the task *marvelcinematicuniverse - marvel* for example, where two mappings $< marvelcinematicuniverse : Zodiac,\ marvel : Zodiac >$ and $< marvelcinematicuniverse : Zodiac,\ marvel : Zodiac\_Virus >$ are generated by our system and eventually the former is selected whereas the latter is the correct match in gold standard. None whatsoever relevant structural information within the two

Table 7. Results for *Large BioMedical Ontologies* track

Matching Task		Size	Prec.	Rec.	F-m.
FMA-NCI	small fragments	2,508	0.967	0.817	0.886
	whole ontologies	3,765	0.622	0.817	0.706
FMA-SNOMED	small fragments	1,720	0.973	0.222	0.362
	FMA whole w/ SNOMED large fragment	1,863	0.881	0.222	0.355
SNOMED-NCI	small fragments	10,910	0.937	0.555	0.697
	SNOMED large fragment w/ NCI whole	12,813	0.789	0.555	0.652

Table 8. Results for *Disease and Phenotype* track

Matching Task	Size	Precision	Recall	F-measure
HP-MP	734	0.997	0.322	0.487
DOID-ORDO	1,274	0.999	0.443	0.614

KGs makes it difficult to do the right decision. For such cases, external resources shall be exploited, providing necessary knowledge for the domain of interest.

3.3 Comments on the OAEI procedure

With respect to the OAEI procedure, the Knowledge Graph track that our system participated in this year is adequately well designed, with organizers being very supportive in resolving issues arisen in the middle of execution phase. The only difficulty we encountered comes from a dependency on Jena packages on the SEALS platform. The problem got settled successfully in the end, and it might be helpful if participants whose systems include Jena packages can be informed in advance that re-packaging Jena on SEALS platform requires additional declaration of the Global Location Mapper. Overall, we sincerely appreciate the efforts by organizers in establishing the OAEI campaign, and with the prospect of further improving the system, we look forward to be back next year.

4 Conclusions

In this paper, we present a variant of FCA-Map called FCAMap-KG, which is particularly designed for matching knowledge graphs. KGs are normally of large size and focus on describing instance connected with properties rather than schematic knowledge of classes as in domain ontologies. FCAMap-KG's performance in the OAEI 2019 *Knowledge Graph* track, together with its two predecessors, demonstrates the power of our FCA-based approach in detecting correspondences across different kinds of data and knowledge representation systems. With the prevail of knowledge graph research in Semantic Web and knowledge engineering community and in industry, extending our system with comprehensive functions and frameworks shall contribute more to this thriving domain.

Table 9. Results for *Biodiversity and Ecology* track

Matching Task	Size	Precision	Recall	F-measure
FLOPO-PTO	171	0.836	0.601	0.699
ENVO-SWEET	422	0.803	0.518	0.630

Acknowledgements

This work was done when the first author was an intern at the Academy of Mathematics and Systems Science, Chinese Academy of Sciences. The work has been supported by the National Key Research and Development Program of China under grant 2016YFB1000902 and the Natural Science Foundation of China grant 61621003.

References

1. Berry, A., Gutierrez, A., Huchard, M., Napoli, A., Sigayret, A.: Hermes: a simple and efficient algorithm for building the AOC-poset of a binary relation. Annals of Mathematics and Artificial Intelligence **72**(1-2), 45–71 (2014)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
3. Chen, G., Zhang, S.: FCAMapX results for OAEI 2018. In: OM@ ISWC. pp. 160–166 (2018)
4. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer Science & Business Media (2012)
5. Hertling, S., Paulheim, H.: DBkWik: A Consolidated Knowledge Graph from Thousands of Wikis. In: 2018 IEEE International Conference on Big Knowledge (ICBK). pp. 17–24. IEEE (2018)
6. Hofmann, A., Perchani, S., Portisch, J., Hertling, S., Paulheim, H.: DBkWik: Towards Knowledge Graph Creation from Thousands of Wikis. In: International Semantic Web Conference (Posters, Demos & Industry Tracks) (2017)
7. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: International Conference on Formal Concept Analysis. pp. 314–339. Springer (2009)
8. Zhao, M., Zhang, S.: FCA-Map results for OAEI 2016. In: OM@ ISWC. pp. 172–177 (2016)
9. Zhao, M., Zhang, S.: Identifying and validating ontology mappings by formal concept analysis. In: OM@ ISWC. pp. 61–72 (2016)
10. Zhao, M., Zhang, S., Li, W., Chen, G.: Matching biomedical ontologies based on formal concept analysis. Journal of biomedical semantics **9**(1), 11 (2018)

FTRLIM Results for OAEI 2019 ⋆ ⋆⋆

Xiaowen Wang[1], Yizhi Jiang[1], Yi Luo[1], Hongfei Fan[1], Hua Jiang[1], Hongming Zhu[1]⋆⋆⋆, and Qin Liu[1,2]

[1] School of Software Engineering, Tongji University, Shanghai, China
[2] Tsingtao Advanced Research Institute, Tongji University, Shanghai, China
{1931533,1931566,1731530,1653545,fanhongfei,
zhu_hongming,qin.liu}@tongji.edu.cn

Abstract. To achieve better efficiency and feasibility in instance matching between two datasets, we proposed a system named FTRLIM, which is based on the FTRL (Follow the Regularized Leader) model. The FTRLIM system supports the generation of indexes for instances, which enables the system to figure out possible matching instance pairs efficiently. FTRLIM participated in the SPIMBENCH track of OAEI 2019, and obtained the highest F-measure in SANDBOX and almost the highest F-measure in MAINBOX, with the least time cost. The results also provided potential directions for further improvement of FTRLIM.

1 Presentation of the system

1.1 State, purpose, general statement

Researchers have worked a lot on ontology alignment, and early methods mainly focused on matching ontologies based on the schema. Recently, the instance-based matching has gradually become a promising topic.[1] There exists many ontology matching systems that support the solution of the instance matching problem, such as LogMap[2], AML[3], Lily[4], RiMOM-IM[5] and so on. With the rapid growth of data scale, it has become a practical requirement to complete the task of instance matching among large-scale knowledge graphs.

FTRLIM is designed to provide an effective and efficient solution for matching instances among large-scale datasets, whose core functionalities are listed as follows:

1. Build indexes for instances based on textual attributes. Only instances with the same index have the possibility to be aligned.

⋆⋆ This research has been supported by the National Key R&D Program of China (No. 2018YFB0505000), the Science and Technology Commission of Shanghai Municipality (No. 17511107303, No. 17511110202), the National Natural Science Foundation of China (No. 61702374), the Shanghai Sailing Program (No. 17YF1420500) and the Fundamental Research Funds for the Central Universities.

⋆⋆⋆ Corresponding author, email: zhu_hongming@tongji.edu.cn

2. Calculate the similarity between two instances on certain attributes and relationships. Different methods have been used to calculate the similarity according to the data types of attributes or relationships.
3. Generate the train dataset for the FTRL model [6] from the given data automatically. Specific instance pairs are selected as train set during the matching process without manual operations.
4. Aggregate similarities of different attributes and relationships into a similarity score with the FTRL model, which is trained after the generation of the train set.
5. Select aligned instances according to similarity scores between each instance pairs.
6. Customize all procedures based on configuration files.

FTRLIM is a newly developed system and it is the first time that we have participated in the OAEI evaluation. We expect to check the feasibility and efficiency of our system, and thus we rebuilt our system using Java with core functionalities. The complete version of FTRLIM has been developed and deployed on a Spark cluster, which provides the system with ability to deal with large-scale data. The user feedback mechanism has been integrated into the system as well. The system will correct matching results on the basis of feedback. Last but not least, the system also supports merging aligned instances' attributes and relationships.

1.2 Specific techniques used

FTRLIM consists of five major components: Index Generator, Comparator, Train set Generator, Model Trainer and Matcher. The system accepts input instances in OWL format, which are stored in source dataset and target dataset respectively. FTRLIM will find aligned instances between the two datasets. The architecture of FTRLIM is presented in Fig.1.

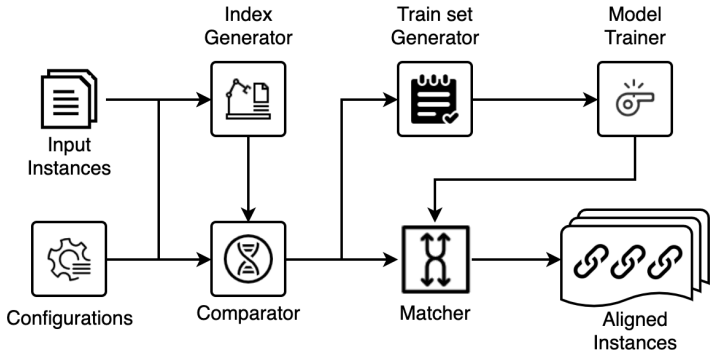


Fig. 1. FTRLIM System OAEI 2019

Index Generator Since the scale of instances that need to be aligned is usually very large, it is very time-consuming and space-consuming to compare all the instances with each other to find aligned instance pairs. FTRLIM uses textual information related to instances to filter out instance pairs that could be aligned efficiently. This work is done by Index Generator. Index Generator plays an important role in FTRLIM. It builds indexes for all input instances based on their attributes. The system first extracts values of a specified instance attribute, then regards each of the values as a document, all of which will constitute a document set. The measurement TF-IDF is used to find keywords for each document. Finally the indexes of an instance are generated from the combination of its keywords. FTRLIM supports users to generate indexes for instances via more than one attribute. In this scenario, different indexes of an instance created referring to different attributes will be concatenated together as the final index. Instances with the same index are divided into the same instance block, and instances from different sources under the same block will form candidate instance pairs. Only when a pair of instances is a candidate pair can it be aligned in the following procedures. When there are only two instances from different data sources in the same block, these two instances will form a unique instance pair[5], which will be regarded as an aligned instance pair directly. Missing value of attributes is taken into consideration to avoid losing candidate instances as far as possible.

Comparator All candidate pairs will be sent to the comparator to calculate similarity. The comparator compares two instances from different aspects. The edit distance similarity is calculated for textual instance attributes, while the Jaccard similarity is calculated for instance relationships. The calculation results will be arranged in order to form the similarity vector. For example, if we compare a candidate pair (x_1, x_2) under two attributes (a_1, a_2) and relationship r_1, the similarities of (x_1, x_2) from each aspect are 0.3, 1 and 0.8, respectively, the similarity vector should be $\langle 0.3, 1, 0.8 \rangle$. All the pairs are compared from identical aspects to ensure that the same dimension of different similarity vectors has the same meaning.

Train set Generator Judging whether a pair of instances is aligned is actually a binary classification problem. We innovatively introduced the FTRL model to solve this problem. The FTRL model has ability to complete the task of classification in large-scale datasets. The model needs to be trained at first. The component, Train set Generator, will generate train set for the FTRL model. The train set is composed of instance pairs' similarity vectors as well as their similarity scores. The Train set Generator regards all unique pairs as aligned pairs. Therefore, it selects all similarity vectors of unique pairs as positive samples, and assigns them with similarity score 1.0. The unaligned pairs are built by replacing one instance of each unique pair randomly. These pairs are assigned with similarity score 0.0 and treated as negative samples in the train set. The input of the FTRL model is the similarity vector, and the output is the similarity

score. This component is different from the complete version of FTRLIM, which will be introduced in Section 1.3.

Model Trainer The FTRL model is trained in this component with hyperparameters in configuration files. Benefiting from the FTRL model's feature, the training process won't cost a long time. The trainer plays a greater role in the complete version as well: it can be used to accept the feedback of users and adjust the parameters of the FTRL model. Users are allowed to choose a batch of candidate instance pairs and correct the similarity score, or pick up a certain pair to correct.

Matcher All candidate pairs will obtain their final similarity scores in this component. The trained FTRL model accepts all the similarity vectors and predicts the matching scores of them. Instance pairs with score larger than 0.5 will be regarded as aligned pairs. They will form the final output of aligned instances together with unique pairs.

Configurations FTRLIM is easily to be tailored according to user's requirements. We expect that all matching procedures are under user's control, thus we allow users to customize their own FTRLIM system using configuration files. Users are able to set the attributes for index generation, the attributes and relationships for comparison, the hyperparameters for the FTRL model and many other detailed parameters to get a better result.

1.3 Adaptions made for the evaluation

To participate in the evaluation, we rebuilt the FTRLIM system and replaced some manual operations with automatic strategies. In the complete version, FTRLIM does not regard all unique pairs as aligned pairs directly. It will compute the mean value of similarity vectors' elements as the raw score for each instance pairs. Then it will select a batch of instance pairs that have raw scores higher than a threshold as positive samples, as well as the same amount of instance pairs whose raw scores are lower than the threshold as negative samples. Users will determine the similarity score by themselves to generate the train set. In the version developed for OAEI, this procedure is changed as we mentioned in 1.2. We excluded the non-core functionalities of the system, and made the ways of input and output suitable for the evaluation.

1.4 Link to the system and parameters file

The implementation of FTRLIM and relevant System Adapter for HOBBIT platform can be found at this FTRLIM-HOBBIT's gitlab page.[3]

[3] https://git.project-hobbit.eu/937522035/ftrlimhobbit

2 Result

In this section, we present the results obtained by FTRLIM in the OAEI 2019 competition. FTRLIM participated in the SPIMBENCH track, which aims at determining when two OWL instances describe the same Creative Work. The datasets are generated and transformed using SPIMBENCH[7]. We are the latest team to join this track. Our competitors are LogMap[2], AML[3] and Lily[4], who have participated in this track for many years. The results are published in this OAEI 2019 result page[4].

2.1 SPIMBENCH

The SPIMBENCH task is executed in two datasets, the SANDBOX and the MAINBOX, of different size. The SANDBOX has about 380 instances and 10000 triplets, while the MAINBOX has about 1800 Create Works and 50000 triplets.

Table 1. The result of SANDBOX

	FTRL-IM	AML	Lily	LogMap
Fmeasure	**0.9214175655**	0.864516129	0.9185867896	0.8413284133
Precision	0.8542857143	0.8348909657	0.8494318182	**0.9382716049**
Recall	**1**	0.8963210702	**1**	0.762541806
Time performance	**1474**	6223	2032	6919

Evaluation results of SANDBOX are summarized in Table 1, where the best results are indicated in bold. Compared with AML[3], Lily[4] and LogMap [2], FTRLIM obtained the highest F-measure, highest recall and best time performance, while the precision is 0.08 lower than LogMap that has the best precision.

Evaluation results of MAINBOX are presented in Table 2 with the best results in bold. Our system is approximately 41% faster than Lily and 17 times faster than the slowest one, while the F-measure is only 0.00014 lower than the best one. We obtained the nearly full mark on recall and the second highest precision as well.

Table 2. The result of MAINBOX

	FTRL-IM	AML	Lily	LogMap
Fmeasure	0.9214787657	0.8604576217	**0.9216224459**	0.790560472
Precision	0.85584563	0.8385678392	0.854638009	**0.8925895087**
Recall	0.9980145599	0.8835208471	**1**	0.7094639312
Time performance	**2155**	39515	3667	26920

[4] http://oaei.ontologymatching.org/2019/results

3 General comments

3.1 Comments on the result

FTRLIM has achieved satisfactory performance in both datasets of SPIMBENCH, especially in the SANDBOX. The Index Generator makes a significant contribution to achieving the results. It helps the system filter out instance pairs with a high possibility to be aligned effectively and efficiently. The comparator only needs to compare instances with the same indexes rather than every instance pairs. The datasets of SPIMBENCH contain a wealth of textual information, and there are many attributes that can be used to build indexes or to compare the similarity among instances. The FTRL model trained by the Model Trainer component is as smart as we expect to learn a weight for attributes or relationships and distinguish pairs of instances pointing to the same entity in real world.

Compared with LogMap, the F-measure of FTRLIM is 8-13% higher while the precision is 4-8% lower. This result shows that FTRLIM could still be improved to obtain higher precision. The OAEI version of FTRLIM considers unique pairs as aligned instances unconditionally, which is not always true. One possible way to solve the problem is validating the matching results. This is one of the centers of our future work.

3.2 Improvements

There are still many aspects to be improved in the FTRLIM system. Besides adding validation stage that described in 3.1, we will continue to optimize the algorithm of generating indexes for instances and the matching strategy in following work. More comparison methods and supporting data types should be attached to our system as well. And we are committed to building the GUI for our system. Although FTRLIM is specially designed to solve the instance matching problem, it is also expected to produce meaningful results in other similar tracks in the future.

4 Conclusion

In this paper, we briefly presented our instance matching system FTRLIM. The core functionalities and components of the system were introduced, and the evaluation results of FTRLIM were presented and analyzed. FTRLIM achieved significantly better time performance than other systems in both datasets of SPIMBENCH, and got the highest F-measure in SANDBOX and almost the same F-measure as the best one in MAINBOX. The results proved the effectiveness and high efficiency of our matching strategy, which is important for matching instances among large-scale datasets.

References

1. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: A literature review. Expert Systems with Applications **42**(2), 949–971 (2015)
2. Jiménez-Ruiz, E., Grau, B.C., Cross, V.V.: Logmap family participation in the oaei 2018. In: OM@ISWC (2018)
3. Faria, D., Pesquita, C., Balasubramani, B.S., Tervo, T., Carriço, D., Garrilha, R., Couto, F.M., Cruz, I.F.: Results of aml participation in oaei 2018. In: OM@ISWC (2018)
4. Tang, Y., Wang, P., Pan, Z., Liu, H.: Lily results for oaei 2018. In: OM@ISWC (2018)
5. Shao, C., Hu, L., Li, J.Z., Wang, Z., Chung, T.L., Xia, J.B.: Rimom-im: A novel iterative framework for instance matching. Journal of Computer Science and Technology **31**, 185–197 (2016)
6. McMahan, H.B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., Chikkerur, S., Liu, D., Wattenberg, M., Hrafnkelsson, A.M., Boulos, T., Kubica, J.: Ad click prediction: a view from the trenches. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2013)
7. Saveta, T., Daskalaki, E., Flouris, G., Fundulaki, I., Herschel, M., Ngomo, A.C.N.: Spimbench : A scalable , schema-aware instance matching benchmark for the semantic publishing domain (2014)

Lily Results for OAEI 2019*

Jiangheng Wu, Zhe Pan, Ce Zhang, Peng Wang

School of Computer Science and Engineering, Southeast University, China
{jiangh_wu, pwang} @ seu.edu.cn

Abstract. This paper presents the results of Lily in the ontology alignment contest OAEI 2019. As a comprehensive ontology matching system, Lily is intended to participate in three tracks of the contest: anatomy, conference, and spimbench. The specific techniques used by Lily will be introduced briefly. The strengths and weaknesses of Lily will also be discussed.

1 Presentation of the system

With the use of hybrid matching strategies, Lily, as an ontology matching system, is capable of solving some issues related to heterogeneous ontologies. It can process normal ontologies, weak informative ontologies [1], ontology mapping debugging [2], and ontology matching tunning [3], in both normal and large scales. In previous OAEI contests [4–10], Lily has achieved preferable performances in some tasks, which indicated its effectiveness and wideness of availability.

1.1 State, purpose, general statement

The core principle of matching strategies of Lily is utilizing the useful information correctly and effectively. Lily combines several effective and efficient matching techniques to facilitate alignments. There are five main matching strategies: (1) Generic Ontology Matching (GOM) is used for common matching tasks with normal size ontologies. (2) Large scale Ontology Matching (LOM) is used for the matching tasks with large size ontologies. (3) Instance Ontology Matching (IOM) is used for instance matching tasks. (4) Ontology mapping debugging is used to verify and improve the alignment results. (5) Ontology matching tuning is used to enhance overall performance.

The matching process mainly contains three steps: (1) Pre-processing, when Lily parses ontologies and prepares the necessary information for subsequent steps. Meanwhile, the ontologies will be generally analyzed, whose characteristics, along with studied datasets, will be utilized to determine parameters and strategies. (2) Similarity computing, when Lily uses special methods to calculate

* This work is supported by National Key R&D Program of China (2018YFD1100302).

the similarities between elements from different ontologies. (3) Post-processing, when alignments are extracted and refined by mapping debugging.

In this year, some algorithms and matching strategies of Lily have been modified for higher efficiency, and adjusted for brand-new matching tasks like Author Recognition and Author Disambiguation in the Instance Matching track.

1.2 Specific techniques used

Lily aims to provide high quality 1:1 concept pair or property pair alignments. The main specific techniques used by Lily are as follows.

Semantic subgraph An element may have heterogeneous semantic interpretations in different ontologies. Therefore, understanding the real local meanings of elements is very useful for similarity computation, which are the foundations for many applications including ontology matching. Therefore, before similarity computation, Lily first describes the meaning for each entity accurately. However, since different ontologies have different preferences to describe their elements, obtaining the semantic context of an element is an open problem. The semantic subgraph was proposed to capture the real meanings of ontology elements [11]. To extract the semantic subgraphs, a hybrid ontology graph is used to represent the semantic relations between elements. An extracting algorithm based on an electrical circuit model is then used with new conductivity calculation rules to improve the quality of the semantic subgraphs. It has been shown that the semantic subgraphs can properly capture the local meanings of elements [11].

Based on the extracted semantic subgraphs, more credible matching clues can be discovered, which help reduce the negative effects of the matching uncertainty.

Generic ontology matching method The similarity computation is based on the semantic subgraphs, which means all the information used in the similarity computation comes from the semantic subgraphs. Lily combines the text matching and structure matching techniques.

Semantic Description Document (SDD) matcher measures the literal similarity between ontologies. A semantic description document of a concept contains the information about class hierarchies, related properties and instances. A semantic description document of a property contains the information about hierarchies, domains, ranges, restrictions and related instances. For the descriptions from different entities, the similarities of the corresponding parts will be calculated. Finally, all separated similarities will be combined with the experiential weights.

Matching weak informative ontologies Most existing ontology matching methods are based on the linguistic information. However, some ontologies may lack in regular linguistic information such as natural words and comments. Consequently the linguistic-based methods will not work. Structure-based methods

are more practical for such situations. Similarity propagation is a feasible idea to realize the structure-based matching. But traditional propagation strategies do not take into consideration the ontology features and will be faced with effectiveness and performance problems. Having analyzed the classical similarity propagation algorithm, *Similarity Flood*, we proposed a new structure-based ontology matching method [1]. This method has two features: (1) It has more strict but reasonable propagation conditions which lead to more efficient matching processes and better alignments. (2) A series of propagation strategies are used to improve the matching quality. We have demonstrated that this method performs well on the OAEI benchmark dataset [1].

However, the similarity propagation is not always perfect. When more alignments are discovered, more incorrect alignments would also be introduced by the similarity propagation. So Lily also uses a strategy to determine when to use the similarity propagation.

Large scale ontology matching Matching large ontologies is a challenge due to its significant time complexity. We proposed a new matching method for large ontologies based on reduction anchors [12]. This method has a distinct advantage over the divide-and-conquer methods because it does not need to partition large ontologies. In particular, two kinds of reduction anchors, positive and negative reduction anchors, are proposed to reduce the time complexity in matching. Positive reduction anchors use the concept hierarchy to predict the ignorable similarity calculations. Negative reduction anchors use the locality of matching to predict the ignorable similarity calculations. Our experimental results on the real world datasets show that the proposed methods are efficient in matching large ontologies [12].

Ontology mapping debugging Lily utilizes a technique named *ontology mapping debugging* to improve the alignment results [2]. Different from existing methods that focus on finding efficient and effective solutions for the ontology mapping problems, mapping debugging emphasizes on analyzing the mapping results to detect or diagnose the mapping defects. During debugging, some types of mapping errors, such as redundant and inconsistent mappings, can be detected. Some warnings, including imprecise mappings or abnormal mappings, are also locked by analyzing the features of mapping result. More importantly, some errors and warnings can be repaired automatically or can be presented to users with revising suggestions.

Ontology matching tuning Lily adopted ontology matching tuning this year. By performing parameter optimization on training datasets [3], Lily is able to determine the best parameters for similar tasks. Those data will be stored. When it comes to real matching tasks, Lily will perform statistical calculations on the new ontologies to acquire their features that help it find the most suitable configurations, based on previous training data. In this way, the overall performance can be improved.

Currently, ontology matching tuning is not totally automatic. It is difficult to find out typical statistical parameters that distinguish each task from others. Meanwhile, learning from test datasets can be really time-consuming. Our experiment is just a beginning.

1.3 Adaptations made for the evaluation

For anatomy and conference tasks, Lily is totally automatic, which means Lily can be invoked directly from the SEALS client. It will also determine which strategy to use and the corresponding parameters. For a specific instance matching task, Lily needs to be configured and started up manually, so only matching results were submitted.

2 Results

2.1 Anatomy track

The anatomy matching task consists of two real large-scale biological ontologies. Table 1 shows the performance of Lily in the Anatomy track on a server with one 3.46 GHz, 6-core CPU and 8GB RAM allocated. The time unit is second (s).

Table 1. The performance in the Anatomy track

Matcher	Runtime	Precision	Recall	Recall+	F-Measure
Lily	281	0.873	0.796	0.52	0.833

Compared with the result in OAEI 2018 [4], there is a sight improvement in Precision, Recall and F-Measure. However, as can be seen in the overall result, Lily lies in the middle position of the rank, which indicates it is still possible to make further progress. External knowledge will be leveraged in the future for the better results. Additionally, to futher reduce the time consumption, some key algorithms will be parallelized.

2.2 Conference track

In this track, there are 7 independent ontologies that can be matched with one another. The 21 subtasks are based on given reference alignments. As a result of heterogeneous characters, it is a challenge to generate high-quality alignments for all ontology pairs in this track.

Lily adopted ontology matching tuning for the Conference track this year. Table 2 shows its latest performance.

Table 2. The performance in the Conference track

Test Case ID	Precision	Recall	F.5-Measure	F1-measure	F2-measure
ra1-M1	0.59	0.6	0.61	0.62	0.63
ra1-M3	0.59	0.58	0.56	0.54	0.53
ra2-M1	0.58	0.58	0.57	0.56	0.56
ra2-M3	0.58	0.56	0.53	0.50	0.48
rar2-M1	0.60	0.59	0.57	0.55	0.44
rar2-M3	0.54	0.53	0.52	0.51	0.50
Average	**0.58**	**0.57**	**0.56**	**0.55**	**0.52**

Compared with the result in OAEI 2018 [4], there is no obvious progress in mean Precision, Recall and F-Measure. All the tasks share the same configurations, so it is possible to generate better alignments by assigning the most suitable parameters for each task. The performance of Lily was even worse than StringEquiv in some tasks. 'We will further analyze this task and our system to find out the reason later.

2.3 Spimbench track

This tack is an instance-mactching tack which aims to match instances of creative works between two boxes. And ontology instances are described through 22 classes, 31 DatatypeProperty and 85 ObjectProperty properties.

There are about 380 instances and 10000 triples in sandbox, and about 1800 CWs and 50000 triples in mainbox.

Table 3. Performance in the spimbench task

Track	Matcher	Precision	Recall	F-Measure	Time
SANDBOX	AML	0.8349	0.8963	0.8645	6223
	FTRL-IM	0.8543	**1.000**	**0.9214**	**1474**
	LogMap	**0.9383**	0.7625	0.8413	6919
	Lily	0.8494	**1.000**	0.9186	2032
MAINBOX	AML	0.8386	0.8835	0.8605	39515
	FTRL-IM	0.8558	0.9980	0.9215	**2155**
	LogMap	**0.8926**	0.7095	0.7906	26920
	Lily	0.8546	**1.000**	**0.9216**	3667

Lily utilized almost the same startegy to handle these two different size tasks. We found that creative works in this task was rich in text information such as titles, descriptions and so on. However, garbled texts and messy codes were mixed up with normal texts. And Lily relied too much on text similarity calculation and set a low threshold in this task, which accounted for the low precision.

As is shown in Table 3, Lily outperforms the others in mainbox. And we suppose that Lily and FTRL-IM share similar strategies in this track as their results are close. Meanwhile, experiments shows that simple ensemble methods and a low threshold contribute to increase of matching efficiency. Nevertheless, compared with FTRL-IM, there is still potential for Lily to speed up in process of matching.

3 General comments

In this year, a lot of modifications were done to Lily for both effectiveness and efficiency. The performance has been improved as we have expected. The strategies for new tasks have been proved to be useful.

On the whole, Lily is a comprehensive ontology matching system with the ability to handle multiple types of ontology matching tasks, of which the results are generally competitive. However, Lily still lacks in strategies for some newly developed matching tasks. The relatively high time and memory consumption also prevent Lily from finishing some challenging tasks.

4 Conclusion

In this paper, we briefly introduced our ontology matching system Lily. The matching process and the special techniques used by Lily were presented, and the alignment results were carefully analyzed.

There is still so much to do to make further progress. Lily needs more optimization to handle large ontologies with limited time and memory. Thus, techniques like parallelization will be applied more. Also, we have just tried out ontology matching tuning. With further research on that, Lily will not only produce better alignments for tracks it was intended for, but also be able to participate in the interactive track.

References

1. Wang, P., Xu, B.: An effective similarity propagation model for matching ontologies without sufficient or regular linguistic information. In: The 4th Asian Semantic Web Conference (ASWC2009), Shanghai, China (2009)
2. Wang, P., Xu, B.: Debugging ontology mappings: a static approach. Computing and Informatics **27**(1), 21–36 (2012)
3. Yang, P., Wang, P., Ji, L., Chen, X., Huang, K., Yu, B.: Ontology matching tuning based on particle swarm optimization: Preliminary results. In: Chinese Semantic Web and Web Science Conference. pp. 146–155. Springer (2014)
4. Tang, Y., Wang, P., Pan, Z., Liu, H.: Lily results for OAEI 2018. In: Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. pp. 179–186 (2018)

5. Wang, P., Wang, W.: Lily results for OAEI 2016. In: Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. pp. 178–184 (2016)
6. Wang, W., Wang, P.: Lily results for OAEI 2015. In: Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. pp. 162–170 (2015)
7. Wang, P.: Lily results on SEALS platform for OAEI 2011. In: Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011 (2011)
8. Wang, P., Xu, B.: Lily: Ontology alignment results for OAEI 2009. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009 (2009)
9. Wang, P., Xu, B.: Lily: Ontology alignment results for OAEI 2008. In: Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008 (2008)
10. Wang, P., Xu, B.: LILY: the results for the ontology alignment contest OAEI 2007. In: Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007 (2007)
11. Wang, P., Xu, B., Zhou, Y.: Extracting semantic subgraphs to capture the real meanings of ontology elements. Tsinghua Science and Technology **15**(6), 724–733 (2010)
12. Wang, P., Zhou, Y., Xu, B.: Matching large ontologies based on reduction anchors. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. pp. 2343–2348 (2011)

LogMap Family Participation in the OAEI 2019 *

Ernesto Jiménez-Ruiz[1,2]

[1] Department of Computer Science, City, University of London, UK
[2] Department of Informatics, University of Oslo, Oslo, Norway

Abstract. We present the participation of LogMap and its variants in the OAEI 2019 campaign. The LogMap project started in January 2011 with the objective of developing a scalable and logic-based ontology matching system. This is the ninth participation in the OAEI and the experience has so far been very positive. LogMap is one of the few systems that participates in (almost) all OAEI tracks.

1 Presentation of the system

LogMap [11, 13] is a highly scalable ontology matching system that implements the consistency and locality principles [12]. LogMap is one of the few ontology matching system that *(i)* can efficiently match semantically rich ontologies containing tens (and even hundreds) of thousands of classes, *(ii)* incorporates sophisticated reasoning and repair techniques to minimise the number of logical inconsistencies, and *(iii)* provides support for user intervention during the matching process.

LogMap relies on the following elements, which are keys to its favourable scalability behaviour (see [11, 13] for details).

Lexical indexation. An inverted index is used to store the lexical information contained in the input ontologies. This index is the key to efficiently computing an initial set of mappings of manageable size. Similar indexes have been successfully used in information retrieval and search engine technologies [2].

Logic-based module extraction. The practical feasibility of unsatisfiability detection and repair critically depends on the size of the input ontologies. To reduce the size of the problem, we exploit ontology modularisation techniques. Ontology modules with well-understood semantic properties can be efficiently computed and are typically much smaller than the input ontology (e.g. [5]).

Propositional Horn reasoning. The relevant modules in the input ontologies together with (a subset of) the candidate mappings are encoded in LogMap using a Horn propositional representation. Furthermore, LogMap implements the classic Dowling-Gallier algorithm for propositional Horn satisfiability [6]. Such encoding, although incomplete, allows LogMap to detect unsatisfiable classes soundly and efficiently.

Axiom tracking. LogMap extends Dowling-Gallier's algorithm to track all mappings that may be involved in the unsatisfiability of a class. This extension is key to implementing a highly scalable repair algorithm.

Local repair. LogMap performs a greedy local repair; that is, it repairs unsatisfiabilities on-the-fly and only looks for the first available repair plan.

Semantic indexation. The Horn propositional representation of the ontology modules and the mappings is efficiently indexed using an interval labelling schema [1] — an optimised data structure for storing directed acyclic graphs (DAGs) that significantly reduces the cost of answering taxonomic queries [4, 16]. In particular, this semantic index allows us to answer many entailment queries as an index lookup operation over the input ontologies and the mappings computed thus far, and hence without the need for reasoning. The semantic index complements the use of the propositional encoding to detect and repair unsatisfiable classes.

1.1 LogMap variants in the 2019 campaign

As in previous campaigns, in the OAEI 2019 we have participated with two additional variants:

LogMapLt is a "lightweight" variant of LogMap, which essentially only applies (efficient) string matching techniques.

LogMapBio includes an extension to use BioPortal [8, 9] as a (dynamic) provider of mediating ontologies instead of relying on a few preselected ontologies [3].

In previous years we also participated with LogMapC[3].

1.2 Link to the system and parameters file

LogMap is open-source and released under GNU Lesser General Public License 3.0.[4] LogMap components and source code are available from the LogMap's GitHub page: `https://github.com/ernestojimenezruiz/logmap-matcher/`.

LogMap distributions can be easily customized through a configuration file containing the matching parameters.

LogMap, including support for interactive ontology matching, can also be used directly through an AJAX-based Web interface: `http://krrwebtools.cs.ox.ac.uk/`. This interface has been very well received by the community since it was deployed in 2012. More than 3,000 requests coming from a broad range of users have been processed so far.

1.3 LogMap as a mapping repair system

Only a very few systems participating in the OAEI competition implement repair techniques. As a result, existing matching systems (even those that typically achieve very high precision scores) compute mappings that lead in many cases to a large number of unsatisfiable classes.

[3] LogMapC is a variant of LogMap which, in addition to the consistency and locality principles, also implements the conservativity principle (see details in [17–19, 15]).

[4] `http://www.gnu.org/licenses/`

We believe that these systems could significantly improve their output if they were to implement repair techniques similar to those available in LogMap. Therefore, with the goal of providing a useful service to the community, we have made LogMap's ontology repair module (LogMap-Repair) available as a self-contained software component that can be seamlessly integrated in most existing ontology matching systems [14, 7].

1.4 LogMap as a matching task division system

LogMap also includes a novel module to divide the ontology alignment task into (independent) manageable subtasks [10]. This component relies on LogMap's lexical index, a neural embedding model [20] and locality-based modules [5]. This module can be integrated in existing ontology alignment systems as a external module. The preliminaty results in [10] are encouraging as the division enabled systems to complete some large-scale matching tasks.

2 General comments and conclusions

Please refer to `http://oaei.ontologymatching.org/2019/results/` for the results of the LogMap family in the OAEI 2019 campaign.

2.1 Comments on the results

As in previous campaigns, LogMap has been one of the top systems and one of the few systems that participates in (almost) all tracks. Furthermore, it has also been one of the few systems implementing repair techniques and providing (almost) coherent mappings in all tracks.

LogMap's main weakness is that the computation of candidate mappings is based on the similarities between the vocabularies of the input ontologies; hence, in the cases where the ontologies are lexically disparate or do not provide enough lexical information LogMap is at a disadvantage.

Acknowledgements

This work was partially supported by the AIDA project, funded by the UK Government's Defence & Security Programme in support of the Alan Turing Institute, and the SIRIUS Centre for Scalable Data Access (Research Council of Norway, project no.: 237889).

I would also like to thank Bernardo Cuenca-Grau, Ian Horrocks, Alessandro Solimando, Valerie Cross, Anton Morant, Yujiao Zhou, Weiguo Xia, Xi Chen, Yuan Gong and Shuo Zhang, who have contributed to the LogMap project in the past.

References

1. Agrawal, R., Borgida, A., Jagadish, H.V.: Efficient management of transitive relationships in large data and knowledge bases. In: ACM SIGMOD Conf. on Management of Data. pp. 253–262 (1989)
2. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)
3. Chen, X., Xia, W., Jiménez-Ruiz, E., Cross, V.: Extending an ontology alignment system with bioportal: a preliminary analysis. In: Poster at Int'l Sem. Web Conf. (ISWC) (2014)
4. Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S.: On labeling schemes for the Semantic Web. In: Int'l World Wide Web (WWW) Conf. pp. 544–555 (2003)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. 31, 273–318 (2008)
6. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Prog. 1(3), 267–284 (1984)
7. Faria, D., Jiménez-Ruiz, E., Pesquita, C., Santos, E., Couto, F.M.: Towards annotating potential incoherences in bioportal mappings. In: 13th Int'l Sem. Web Conf. (ISWC) (2014)
8. Fridman Noy, N., Shah, N.H., Whetzel, P.L., Dai, B., et al.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37, 170–173 (2009)
9. Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N.H., Musen, M.A.: What four million mappings can tell you about two hundred ontologies. In: Int'l Sem. Web Conf. (ISWC) (2009)
10. Jiménez-Ruiz, E., Agibetov, A., Samwald, M., Cross, V.: Breaking-down the ontology alignment task with a lexical index and neural embeddings. CoRR abs/1805.12402 (2018), http://arxiv.org/abs/1805.12402
11. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and Scalable Ontology Matching. In: Int'l Sem. Web Conf. (ISWC). pp. 273–288 (2011)
12. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources. J. Biomed. Sem. 2 (2011)
13. Jiménez-Ruiz, E., Cuenca Grau, B., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: Algorithms and implementation. In: Europ. Conf. on Artif. Intell. (ECAI) (2012)
14. Jiménez-Ruiz, E., Meilicke, C., Cuenca Grau, B., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: 26th Description Logics Workshop (2013)
15. Jimenez-Ruiz, E., Payne, T.R., Solimando, A., Tamma, V.: Limiting logical violations in ontology alignment through negotiation. In: Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR). AAAI Press (April 2016)
16. Nebot, V., Berlanga, R.: Efficient retrieval of ontology fragments using an interval labeling scheme. Inf. Sci. 179(24), 4151–4173 (2009)
17. Solimando, A., Jiménez-Ruiz, E., Guerrini, G.: Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In: Int'l Sem. Web Conf. (ISWC) (2014)
18. Solimando, A., Jiménez-Ruiz, E., Guerrini, G.: A multi-strategy approach for detecting and correcting conservativity principle violations in ontology alignments. In: Proc. of the 11th International Workshop on OWL: Experiences and Directions (OWLED). pp. 13–24 (2014)
19. Solimando, A., Jimenez-Ruiz, E., Guerrini, G.: Minimizing conservativity violations in ontology alignments: Algorithms and evaluation. Knowledge and Information Systems (2016), https://github.com/asolimando/logmap-conservativity/
20. Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., Weston, J.: Starspace: Embed all the things! arXiv preprint arXiv:1709.03856 (2017)

ONTMAT1: Results for OAEI 2019*

Saida Gherbi[1]and M[ed]Tarek Khadir[2]

[1]LabGed, ESTI, Annaba 23000, Algeria
Saida_gharbi23@yahoo.fr
[2]LabGed, University Badji Mokhtar Annaba, 23000, Algeria
Khadir@labged.net

Abstract: This paper describes an ontology matching system named ONTMAT1, and presents the results obtained for the Ontology Alignment Evaluation Initiative (OAEI) 2019. ONTMAT1 compares entities of ontologies to align by structural and terminological methods which use a reasoner along with wordnet dictionnary. Thus, based on similarities of individual, datatype properties and the semantic of property restriction, the weight that estimates the performance of structural and linguistic similarities is calculated.

Keywords: Ontology, Alignment, OWL.

1 Presentation of the system

ONTMAT1 (ONTology MATching) is an ontology alignment tool, aiming to align OWL entities (classes, object properties), participating for the first time in OAEI (Conference track). The specificities of ONTOMAT1 are presented below:

1.1 State, purpose, general statement

ONTMAT1 uses terminological methods based on n-gram measure and WordNet dictionary [1] that is exploited as background knowledge along with pellet reasoner [2], to provide synonyms of names of individuals, concepts, and properties, of ontologies source ($O1$) and target($O2$). The results obtained are saved in: individual matrix (M_{ind}), concepts matrix (M_{cpt}), and properties matrix (M_p), for individuals, concepts and properties, respectively.

Furthermore, a new weight that evaluates the impact of restriction property (object properties [3] and data type properties) on the structural similarity of concept is calculated. Thus, the impact of terminological similarity is 1 minus this weight. Then, the final result of concepts alignment is the sum of these similarities.

1.2 Approach description

The suggested algorithm is composed of 3 levels as explain in the following:

1. In level 1, normalization techniques such as lemmatization [4], are applied on each entities name of matrices (M_{cpt}, M_{ind}, M_p). Then, the n-gram measure is used to assess the similarity among these entities. This measure is opted because it permits the control of the lexicon size and keeping at the same time a reasonable threshold for every composed term (names).The obtained value is assigned to the intersection between entities into every matrix. Since, the metric measures used to align entities may suffer of several drawbacks, such as: the existence of synonyms that expresses the same entity using different words. Entities names are also compared to WordNet synsets using n-gram and the relation among synsets are inferred by Pellet reasoner. Then, the relations among these entities are deduced from relations inferred by the reasoner.

   If *synonym* relation is inferred, then the value of intersection among these entities in their matrix becomes the average between 1.0 and the value calculated by the n-gram measure, else the existent value is preserved.
2. In level 2, every property restriction defines the class allocated by a weight w_i that evaluates the influence of its semantic on this class.

   The sublanguage OWL-DL of OWL (Web Ontology Language) certified by the World Wide Web Consortium (W3C)[i] is adopted in this paper to define the offered ontology matching algorithm. This language distinguishes two types of property restrictions: value constraints and cardinality constraints, which give a semantic sense to the assessed weight. A value constraint applies constraints on the range of the property. These constraints put on the class C or an object *o* can be:

   - *allValuesFrom(C),* is the same to the universal (for-all: ∀) quantifier of Predicate logic that for each instance of C, every value for Property must satisfies the constraint. Therefore, the algorithm can assert that this property has a robust impact on the class. Consequently, from its semantic, the influence of this restriction on the class is considered "strong" and suggested 1.0 as weights w_{1i}, w_{2j} in O_1, O_2, respectively, affected by ONTMAT1 to *allValuesFrom*.
   - *someValuesFrom(C),* is similar to the existential quantifier of Predicate logic that for each instance of C, there exists at least one value for Property that satisfies the constraint. Therefore, the influence of this constraint on a given class can be valued as average and the value 0.75 is affected to w_{1i} in O_1 and w_{2j} in O_2.
   - *hasValue(o),* joins a restriction class to a value *o*, which may be an individual or a data value. This restriction designates a class of all individuals for which the concerned property has at least one value *semantically* equivalent to *o* (it can, also, have supplementary values). The effect of this restriction can be considered as weak and the

assigned weights (w_{1i}, w_{2j} in O_1, O_2, respectively) are evaluated to 0.25.

- A cardinality constraint is defined by *maxCardinality(n)* and *minCardinality(n)*, where (*n*) is the number of values that a property can take. *Owl:maxCardinality(n)* describes a class of all individuals that have *at most n* diverse values (individuals or data values) for the concerned property. The influence of this constraint is only on *n* value, for this reason, it is estimated as a weak constraint and ONTMAT1 affects 0.25 to weights w_{1i}, w_{2j} in O_1, O_2, respectively. The same for *minCardinality(n)* that describes a class of all individuals that have *at least n* various values for the concerned property.

3. Level 3 assesses structural similarity between concepts established upon properties restrictions. Property restrictions can be either *datatype properties* (data literal is the value of properties), or *object properties* (individual is the value of properties)[ii]. Firstly, restriction names of concepts (C_{1i}, C_{2j}) to be matched are compared using terminological methods.
   Secondly, same terminological methods are used to measure similarities among *datatype properties* names of both concepts to align, as well as the average of these similarities is calculated to determine *data similarities*.
   Finally, similarities among individuals of concepts to match are extracted from M_{ind} to compute their average *data similarities*.
   Afterwards, weights w_i and w_j evaluated influences of property on concepts are multiplied by *data similarities* and *data similarities*. Furthermore, values affected to M_p will be replaced by those deduced in this level.
4. The last level consists on aggregation of above similarities of concepts. Consequently, the final similarity is the sum of structural similarity and 1 minus the average of structural weights multiplied by terminological similarity.

1.3 Adaptations made for the evaluation

The alignment format adapted by the results, is the "=" sign for equivalence relation with confidence of 1.

However our system provides other relation called fuzzy relation symbolized by *-1*, proposed to resolve the problem of domination of structural similarity. This relation designates that the suggested system cannot decide about the relation that can be among the entities to match. This relation is assigned to concepts in which the difference between its $TermSim(C_{1i}, C_{2j})$, $StructSim(C_{1i}, C_{2j})$, has a value that exceeds a certain threshold considered according to the expertise of the application in OAEI.

2 Results

In this version we wish to test the techniques used by ONTMAT1, for instance: the inferences mechanisms applied upon WordNet, and the deduction of the matching among entities using weight based on restriction properties. The track used to perform these tests is the conference track. Conference track comprises 16 ontologies from the domain of conference organization.

The results of the evaluation based on crisp reference alignments that contains only classes (*M1-rar2; M1-ra1; M1-ra2*) are considered in this study because the objective of this version is to show the influence of the weight and the reasoner on the classes alignment and properties will be treated in the next version

As depicted in Table 1, ONTMAT1 provides fairly stable alignments when matching conference ontologies. Table 2 illustrates that ONTMAT1's performance in discrete and continuous cases increases 16 percent in terms of F-measure over the sharp reference alignment from 0.55 to 0.64, driven, principally, by increased recall.

Table 1. Results based on the crisp reference alignments.

	Precision	F-Measure 1	Recall
M1-ra1	0.82	*0.61*	0.49
M1-ra2	0.77	0.56	0.44
M1-rar2	0.77	0.58	0.46

Table 2. Results based on the uncertain version of the reference alignment.

Precision	F-measure1	Recall
Uncertain reference alignments (Sharp)		
0.82	0.55	0.41
Uncertain reference alignments (Discrete)		
0.82	0.64	0.52
Uncertain reference alignments (Continuous)		
0.82	0.64	0.53

Finally, ONTMAT1 have generated only one incoherent alignment in the evaluation based on logical reasoning.

2.1 Discussions on the way to improve the proposed system

To improve the proposed application, properties of ontologies ($O1, O2$) will also be aligned. Then, adapt it to read all files type, and integrate the translator to test our tool under other tracks as: Instance Matching, MultiFarm.

3 Conclusion and future work

We have briefly described the mechanisms exploited by our proposition ONTMAT1, and presented the results obtained under the conference track of OAEI 2019.

This is our firs participation in OAEI with ONTMAT1, the results are satisfying, and the system presents some limitations in term of recall. In the future, a greater effort will be made to improve ONTMAT1 results, and participate in more tracks.

References

1. Fellbaum, C. :WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA (1998).
2. Sirin E. et Parsia B. 2004 . "Pellet : An owl dl reasoned", DansHaarslev, V. et Möller, R. (éditeurs), Proceedings of the International Workshop on Description Logics (DL2004).
3. Gherbi S., M. T. Khadir, Inferred ontology concepts alignment using and an external dictionary, Procedia Computer Science 83 (2016) 648- 652, the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Aliated Workshops.doi:http://dx.doi.org/10.1016/j.procs.2016.04.145.URL http://www.sciencedirect.com/science/article/pii/ S1877050916301752.
4. Euzenat, J., Shvaiko. P, (2007). "Ontology Matching" , Springer-Verlag Berlin Heidelberg

[i] https://www.w3.org/

[ii] https://www.w3+.org/TR/owl-ref/

POMap++ Results for OAEI 2019: Fully Automated Machine Learning Approach for Ontology Matching

Amir Laadhar[1], Faiza Ghozzi[2], Imen Megdiche[1], Franck Ravat[1], Olivier Teste[1], and Faiez Gargouri[2]

[1] Paul Sabatier University, IRIT (CNRS/UMR 5505) 118 Route de Narbonne 31062 Toulouse, France
{amir.laadhar,imen.megdiche,franck.ravat,olivier.teste}@irit.fr,
[2] University of Sfax, MIRACL Sakiet Ezzit 3021, Tunisie
{faiza.ghozzi,faiez.gargouri}@isims.usf.tn

Abstract. POMap++ is a novel ontology matching system based on a machine learning approach. This year is the second participation of POMap++ in the Ontology Alignment Evaluation Initiative (OAEI). POMap++ follows a fully automated local matching learning approach that breaks down a large ontology matching task into a set of independent local sub-matching tasks. This approach integrates a novel partitioning algorithm as well as a set of matching learning techniques. POMap++ provides an automated local matching learning for the biomedical tracks. In this paper, we present POMap++ as well as the obtained results for the Ontology Alignment Evaluation Initiative of 2019.

Keywords: Semantic web, Machine learning, ontology matching, ontology partitioning

1 Presentation of the system

1.1 State, purpose, general statement

Ontologies have grown increasingly large in real application domains, notably the biomedical domain, where ontologies, such as the Systematized Nomenclature of Medicine and Clinical Terms (SNOMED CT) with 122464 classes, the National Cancer Institute Thesaurus (NCI) with 150231 classes, and the Foundational Model of Anatomy (FMA) with 104721 classes are widely employed [11]. These ontologies can vastly vary in terms of their modeling standpoints and vocabularies, even for the same domain of interest. To enable interoperability we will need to integrate these large knowledge resources in a single representative resource [1, 3]. This integration can be established through a novel matching process which specifies the correspondences between the entities of heterogeneous ontologies.

Existing ontology matching systems have to overcome two major issues when dealing with large ontologies: (i) integrating the large size not yet feasible with a good matching accuracy, (ii) automating the ontology matching process.

The large size of these ontologies decreases the matching accuracy of ontology matching systems [5]. Large ontologies describing the same domain includes a high conceptual heterogeneity. Ontology developers can construct the same domain ontology but using different conceptual models. As a result, finding mappings between two ontologies became more difficult [9]. Consequently, the matching of large ontologies became error-prone, especially while combining different matchers in order to result in an adequate result [7]. To summarize, the main issues of the alignment of large ontologies are the conceptual heterogeneity, the high search space and the decreased quality of the resulted alignments. Dealing effectively with biomedical ontologies requires a solution that will align large alignment tasks such as "divide and conquer" or parallelization approaches.

While dealing with different matching tasks, the main issue is the automation process is the choice of the matching settings. The matching tuning process should be automated in order to reduce the matching process complexity, especially while dealing with large scale ontologies. As a result, the ontology matching process needs to be self-tuned for a better selection of matching settings for each matching problem. This process can improve the ontology matching accuracy. In the case of large ontologies, it is important to have highly-automated, generic processes which are independent of the input ontologies. To achieve quality alignments, ontology matching systems can employ a variety of matchers while managing complex ontologies. The choice of these matchers should depend on the matching context. In the context of large ontologies, the drawback of manual solutions is the level of complexity and the time needed to generate results for such a large problem.

To respond to the later issues, we propose POMap++ [2, 4, 10] as a novel local matching learning approach that combines ontology partitioning with ontology matching learning. In the following, we briefly describe the main processes of the proposed contributions as depicted in Figure 1. This architectural overviw has two ontologies as the input and alignments as the output. The output is a set of correspondences generated from the two input ontologies.

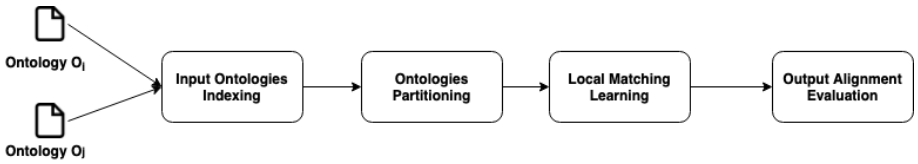


Fig. 1. POMap++ overview

1. The two input ontologies are pre-processed and indexed in the first module. We applied a set of natural language processes across the annotations for

each input ontology. All the annotations and semantic relationships between entities are stored in a data structure.
2. In the second module, the indexed ontologies are then partitioned in order to generate the set of local matching tasks. The partitioning process ensures good coverage of the alignments that should be discovered.
3. In the third module, we automatically build a local classifier for each local matching task. These local classifiers automatically align the set of local matching tasks based on their adequate features.
4. In the fourth module, the generated alignment file stores the set of correspondences located by all the local matching tasks. The correspondences are compared to the reference alignments provided by the Gold Standard to assess the accuracy of local matching.

1.2 Specific techniques used

The workflow of POMap++ for our second participation in the OAEI comprises four main steps, as flagged by the figure 1: Input ontologies indexing and loading, input ontologies partitioning, local matching learning and output alignment generation. The first and the last step are the same as in the last version of POMap++ . In the second step, we define the pair of similar partitions between the two input ontologies. In the third step, we apply machine learning techniques in order to align every identified pair of similar partitions. In the following, we detail the second step and the third step.

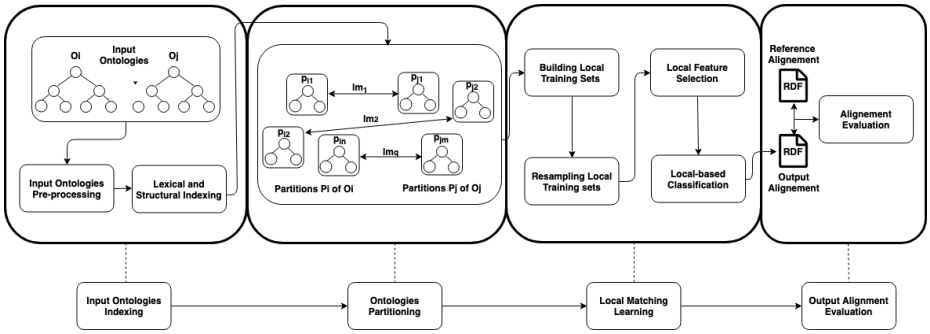


Fig. 2. POMap++ Architecture

Ontologies partitioning [6]: this step is based on a novel partitioning approach based on hierarchical agglomerative clustering. As input, it takes two ontologies and generates as an output a set of local matching tasks. The partitioning approach split a large ontology matching task into a set of sub-matching tasks. The large search space is reduced accordingly to the number of local matching tasks. Therefore, the search space is minimized from the whole ontology matching problem to a set of sub-matching problems. Consequently, the alignment of

the two input ontologies can be more effective for each sub-matching task in order to result in a better matching accuracy for the whole matching problem. The proposed partitioning approach is based on a novel multi-cut strategy generating not large partitions or not isolated ones.

Local matching learning [**8**]: in this step we propose a local matching learning approach in order to fully automate the matching tuning for each local matching task. This automation has to be defined for every new matching context in order to result in a context-independent local matching learning system. This matching system should align each local matching context based on its characteristics. State-of-the-art approaches define a set of predefined matching settings for all the matching contexts. However, the benefit of the local matching learning approach is the use of machine learning methods, which can be flexible and self-configuring during the training process. We apply the proposed matching learning approach locally and not globally. Consequently, we set the adequate matching tuning for each local matching task. Therefore, we result in a better matching quality independently of the matching context. Each local matching task is automatically aligned using its local classifier from its local training set. These local training sets are generated without the use of any reference alignments. Each local classifier automatically defines the matching settings for its local matching task in terms of the appropriate element-level and structural-level matchers, weights and thresholds.

2 Results

2.1 Anatomy

The Anatomy track consists of finding the alignments between the Adult Mouse Anatomy and the NCI Thesaurus describing the human anatomy. The evaluation was run on a server coupled with 3.46 GHz (6 cores) and 8GB of RAM. Table 1 draws the performance of POMap++ compared to the five top matching systems. Our matching system achieved the third best result for this dataset with an F-measure of 89.7%, which is very close to the top results.

Table 1. POMap++ results in the anatomy track compared to the OAEI 2017 systems.

System	Precision	Recall	F-Measure	Runtime
AML	0.95	0.936	0.943	76
LogMapBIo	0.872	0.925	0.898	1718
POMap++	0.919	0.877	0.897	345
LogMap	0.918	0.846	0.880	28
SANOM	0.888	0.844	0.865	516

2.2 Large biomedical ontologies

This tracks aims to find the alignment between three large ontologies: Foundational Model of Anatomy (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI). Among six matching tasks between these three ontologies, POMap++ succeeded to perform the matching between FMA-NCI (small fragments) and FMA-SNOMED (small fragments) with an F-Measure respectively of 88.9% and 40.4%. For the other tasks of the large biomedical track, POMap++ exceeded the defined timeout due to the required time for the training and the generation of machine learning classifiers. As a future work, we are planning to cope with the matching process of the larger ontologies in a shorter time.

2.3 Disease and Phenotype

This track is based on a real use case in order to find alignments between disease and phenotype ontologies. Specifically, the selected ontologies are the Human Phenotype Ontology (HPO), the Mammalian Phenotype Ontology (MP), the Human Disease Ontology (DOID) and the Orphanet and Rare Diseases Ontology(ORDO). The evaluation was run on an Ubuntu Laptop with an Intel Core i7-4600U CPU @ 2.10GHz x 4 coupled with 15Gb RAM. POMap++ produced 1502 mappings in the HP-MP task associated with 218 unique mappings. Among twelve matching systems, POMap++ achieved the fifth highest F-measure with an F-Measure of 83.6%. In the DOID-ORDO task, POMap++ generated 2563 mappings with 192 unique ones. According to the 2-vote silver standard, it scored an F-Measure of 83.6%. We ranked third in the DOID-ORDO task among 8 matching systems

2.4 Biodiversity and Ecology

This track consists on finding alignments between the Environment Ontology (ENVO) and the Semantic Web for Earth and Environment Technology Ontology (SWEET), and between the Flora Phenotype Ontology (FLOPO) and the Plant Trait Ontology (PTO). These ontologies are particularly useful for biodiversity and ecology research and are being used in various projects. They have been developed in parallel and are very overlapping. They are semantically rich and contain tens of thousands of classes. For the FLOPO-PTO matching task, we achieved an F-Measure of 68.1 %. For the FLOPO-PTO matching task, POMap ++ achieved an F-measure of 69.3 %. We ranked as the second best matching system for this task.

3 Conclusion

POMap++ obtained the top results for different matching tasks such as Anatomy, DOID-ORDO and FLOPO-PTO. For the machine learning classifiers, we did not opt to perform the local matching using semantic-level features. Consequently, we are planning to add semantic-level features to the machine learning matching based approach.

References

1. Daniel Faria, Catia Pesquita, Isabela Mott, Catarina Martins, Francisco M Couto, and Isabel F Cruz. 2018. Tackling the challenges of matching biomedical ontologies. Journal of biomedical semantics 9, 1.
2. Laadhar, A., Ghozzi, F., Megdiche Bousarsar, I., Ravat, F., Teste, O., Gargouri, F. (2018). OAEI 2018 results of POMap++. CEUR-WS: Workshop proceedings.
3. Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jrme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Ian Harrow, Valentina Ivanova, et al. 2017. Results of the ontology alignment evaluation initiative 2017. In OM 2017-12th ISWC workshop on ontology matching
4. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F. (2017, October). POMap results for OAEI 2017.
5. Ernesto Jimnez-Ruiz, Asan Agibetov, Matthias Samwald, and Valerie Cross. 2018. We Divide, You Conquer: From Large-scale Ontology Alignment to Manageable Subtasks. (2018).
6. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F. (2019, April). Partitioning and local matching learning of large biomedical ontologies. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 2285-2292). ACM.
7. Xingsi Xue and Jeng-Shyang Pan. 2017. A segment-based approach for large-scale ontology matching. Knowledge and Information Systems 52, 2.
8. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F. (2019, June). The Impact of Imbalanced Training Data on Local Matching Learning of Ontologies. In International Conference on Business Information Systems (pp. 162-175). Springer, Cham.
9. Alsayed Algergawy, Samira Babalou, Mohammad J Kargar, and S Hashem Davarpanah. 2015. Seecont: A new seeding-based clustering approach for ontology matching. In East European Conference on Advances in Databases and Information Systems. Springer.
10. Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F. (2017). POMap: An Effective Pairwise Ontology Matching System. In KEOD (pp. 161-168).
11. Euzenat, Jrme, and Pavel Shvaiko. Ontology matching. Vol. 18. Heidelberg: Springer, 2007.

SANOM Results for OAEI 2019

Majid Mohammadi, Amir Ahooye Atashin, Wout Hofman, and Yao-Hua Tan

Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands,
TNO Research institute, The Netherlands.

Abstract. Simulated annealing-based ontology matching (SANOM) participates for the second time at the ontology alignment evaluation initiative (OAEI) 2019. This paper contains the configuration of SANOM and its results on the anatomy and conference tracks. In comparison to the OAEI 2017, SANOM has improved significantly, and its results are competitive with the state-of-the-art systems. In particular, SANOM has the highest recall rate among the participated systems in the conference track, and is competitive with AML, the best performing system, in terms of F-measure. SANOM is also competitive with LogMap on the anatomy track, which is the best performing system in this track with no usage of particular biomedical background knowledge. SANOM has been adapted to the HOBBIT platfrom and is now available for the registered users. *abstract* environment.

Keywords: SANOM, ontology alignment, OAEI.

1 System Representation

SANOM takes advantages of the well-known simulated annealing (SA) to discover the shared concepts between two given ontologies [3]. A potential alignment is modeled as a state in the SA whose evolution would result in a more reliable matching between ontologies. The evolution requires a fitness function in order to gauge the goodness of the intermediate solutions to the ontology matching problem.

A fitness function should utilize the lexical and structural similarity metrics to estimate the fineness of an alignment. The version of SANOM participated this year uses both lexical and structural similarity metrics, which are described in the following.

1.1 Lexical Similarity Metric

The cleaning of strings before the similarity computation is essential to increase the chance of mapping entities. SANOM uses the following pre-processing techniques to this end:

- **Tokenization.** It is quite common that the The terminology of concepts are constructed from a bag of words (BoW). The words are often concatenated by white space, the capital letter of first letters, and several punctuations such as " − " or " _ ". Therefore, they need to be broken into individual words and then the similarity is computed by comparing the bag of words together.
- **Stop word removal.** Stop words are the typical words with no particular meaning. The stop words should be detected by searching the tokens (identified after tokenization) in a table containing all possible stop words. The Glasgow stop word list is utilized in the current implementation [1] .
- **Stemming.** Two entities from the given ontologies might refer to a similar concept, but they are named differently due to various verb tense, plural/singular, and so forth. Therefore, one needs to recover the normal words so that the similar concepts will have higher similarity. The Porter stemming method is used for this matter [4].

After the pre-processing step, the strings of two concepts can be given to a similarity metric in order to calibrate the degree of similarity between concepts. The base similarity metric computes the sameness of tokens obtained from each entity. The current version of SANOM takes advantage of two similarity metrics and take their maximum as the final similarity of two given tokens. One of this similarity metric is for sole comparison of stirngs, and the other one is to guage the linguistic relation of two given names. These similarity metrics are:

- **Jaro-Winkler metric.** The combination of TF-IDF and Jaro-Winkler is popular and has been sucessful in ontology alignment as well. Similarly, SANOM uses Jaro-Winkler with the threshold 0.9 as one of the base similarity metrics.
- **WordNet-based metric.** The linguistic heterogeneity is also rampant in various domains. Therefore, the existence of a similarity metric to measure the lingual closeness of two entities is absolutely essential. In this study, the relatedness of two given tokens are computed by the Wu and Palmer measure [5] and is used as a base similarity metric with the threshold 0.95.

1.2 Structural Similarity Metric

The preceding string similarity metric gives a high score to the entities which have lexical or linguistic proximity. Another similarity of two entities could be derived from their positions in the given ontologies.

We consider two structural similarity measures for the current implementation of SANOM:

- The first structural similarity is gauged by the subsumption relation of classes. If there are two classes c_1 and c_2 whose superclasses are s_1 and s_2 from two given ontologies O_1 and O_2, then the matching of classes s_1

[1] http://ir.dcs.gla.ac.uk/resources/linguistic utils/stop words

and s_2 would increase the similarity of c_1 and c_2. Let s be a correspondence mapping s_1 to s_2, then the increased similarity of c_1 and c_2 is gauged by

$$f_{structural}(c_1, c_2) = f(s). \tag{1}$$

– Another structural similarity is derived from the properties of the given ontologies. The alignment of two properties would tell us the fact that their corresponding domain and/or ranges are also identical. Similarly, if two properties have the analogous domain and/or range, then it is likely that they are similar as well.
The names of properties and even their corresponding core concepts are not a reliable meter based on which they are declared a correspondence. A recent study has shown that the mapping of properties solely based on their names would result in high false positive and false negative rates, e.g. there are properties with identical names which are not semantically related while there are semantically relevant properties with totally distinct names.
The current implementation treats the object and data properties differently. For the object properties op_1 and op_2, their corresponding domains and ranges are computed as the concatenation of their set of ranges and domains, respectively. Then, the fitness of the names, domains, and ranges are computed by the Soft TF-IDF. The final mapping of two properties is the average of top two fitness scores obtained by the Soft TF-IDF. For the data properties, the fitness is computed as the similarity average of names and their corresponding domain.
On the other flow of alignment, it is possible to derive if two classes are identical based on the properties. Let e_1 and e_2 be classes, op_1 and op_2 be the object properties, and R_1 and R_2 are the corresponding ranges, then the correspondence $c = (e_1, e_2)$ is evaluated as

$$f_{structural}(c) = \frac{f_{string}(R_1, R_2) + f_{string}(op_1, op_2)}{2}. \tag{2}$$

2 Results

This section contains the results obtained by SANOM on the anatomy and conference track.

2.1 Anatomy Track

The anatomy track is one of the earliest benchmarks in the OAEI. The task is about aligning the Adult Mouse anatomy and a part of NCI thesaurus containing the anatomy of humans. Each of the ontologies has approximately 3,000 classes, which are designed carefully and are annotated in technical terms.

The best performing systems in this track use a biomedical background knowledge. Thus, their results are not comparable with SANOM which does not use any particular background knowledge. Among other systems, LogMap [2] is best one with no use of a background knowledge.

Table 1 tabulates the precision, recall, and F-measure of SANOM and LogMap on the anatomy track. According to this table, the recall of SANOM is slightly higher than LogMap which means that it could identify more correspondences than LogMap. However, the precision of LogMap is better than SANOM with the margin of three percent. The overall performance of SANOM is quite close to LogMap since their F-measure has only 1% difference.

System	Precision	F-measure	Recall
LogMap	0.918	0.88	0.846
SANOM	0.888	0.87	0.853

Table 1: The precision, recall, and F-measure of SANOM and LogMap on the OAEI anatomy track.

	SANOM			AML			LogMap		
	P	F	R	P	F	R	P	F	R
cmt-conference	0.61	0.74	0.93	0.67	0.59	0.53	0.73	0.62	0.53
cmt-confOf	0.80	0.62	0.50	0.90	0.69	0.56	0.83	0.45	0.31
cmt-edas	0.63	0.69	0.77	0.90	0.78	0.69	0.89	0.73	0.62
cmt-ekaw	0.54	0.58	0.64	0.75	0.63	0.55	0.75	0.63	0.55
cmt-iasted	0.67	0.80	1.00	0.80	0.89	1.00	0.80	0.89	1.00
cmt-sigkdd	0.85	0.88	0.92	0.92	0.92	0.92	1.00	0.91	0.83
conference-confOf	0.79	0.76	0.73	0.87	0.87	0.87	0.85	0.79	0.73
conference-edas	0.67	0.74	0.82	0.73	0.69	0.65	0.85	0.73	0.65
conference-ekaw	0.66	0.70	0.76	0.78	0.75	0.72	0.63	0.55	0.48
conference-iasted	0.88	0.64	0.50	0.83	0.50	0.36	0.88	0.64	0.50
conference-sigkdd	0.75	0.77	0.80	0.85	0.79	0.73	0.85	0.79	0.73
confOf-edas	0.82	0.78	0.74	0.92	0.71	0.58	0.77	0.63	0.53
confOf-ekaw	0.81	0.83	0.85	0.94	0.86	0.80	0.93	0.80	0.70
confOf-iasted	0.71	0.63	0.56	0.80	0.57	0.44	1.00	0.62	0.44
confOf-sigkdd	0.83	0.77	0.71	1.00	0.92	0.86	1.00	0.83	0.71
edas-ekaw	0.71	0.72	0.74	0.79	0.59	0.48	0.75	0.62	0.52
edas-iasted	0.69	0.56	0.47	0.82	0.60	0.47	0.88	0.52	0.37
edas-sigkdd	0.80	0.64	0.53	1.00	0.80	0.67	0.88	0.61	0.47
ekaw-iasted	0.70	0.70	0.70	0.88	0.78	0.70	0.75	0.67	0.60
ekaw-sigkdd	0.89	0.80	0.73	0.80	0.76	0.73	0.86	0.67	0.55
iasted-sigkdd	0.70	0.80	0.93	0.81	0.84	0.87	0.71	0.69	0.67
Average	0.74	0.72	0.73	0.84	0.74	0.67	0.84	0.68	0.59

Table 2: The precision, recall, and F-measure of SANOM, AML, and LogMap on various datasets on the conference track

2.2 Conference Track

The conference comprises the pairwise alignment of seven ontologies. Table 2 displays the precision, recall, and F-measure of SANOM, LogMap, and AML [1] on the conference track. AML and LogMap are the top two systems in terms of precision and recall.

According to Table 2, the recall of SANOM is superior to both LogMap and AML. SANOM's average recall is 7% and 14% more than those of AML and LogMap, respectively, but its precision is 10% less than both of the systems. Overall, the performance of SANOM is quite competitive with the top performing systems in the conference track.

2.3 Large BioMed Track

The conference comprises the pairwise alignment of seven ontologies. Table 3 displays the precision, recall, and F-measure of SANOM, LogMap, and AML [1] on the Large BioMed track. AML and LogMap are the top two systems in terms of precision and recall.

	SANOM			AML			LogMap		
	P	F	R	P	F	R	P	F	R
FMA-NCI (whole)	0.61	0.74	0.841	0.805	0.59	0.881	0.856	0.831	0.808
FMA-SNOMED (whole)	0.905	0.283	0.167	0.685	0.697	0.710	0.840	0.730	0.645
SNOMED-NCI (whole)	0.868	0.618	0.479	0.862	0.765	0.687	0.867	0.706	0.596

Table 3: The precision, recall, and F-measure of SANOM, AML, and LogMap on various datasets on the Large BioMed track

3 Conclusion

SANOM only participated in the OAEI 2019 anatomy, conference and Large BioMed track. For the next year, we have aims to participate in more tracks so that the performance of SANOM can be compared with that of the state-of-the-art systems in other tracks as well. Another avenue to improve the system is to equip it with a proper biomedical background knowledge since most of the OAEI tracks are from this domain.

References

1. Daniel Faria, Catia Pesquita, Booma Balasubramani, Teemu Tervo, David Carriço, Rodrigo Garrilha, Francisco Couto, and Isabel F Cruz. Results of aml participation in oaei 2018. In *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference*, volume 2288, 2018.

2. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference*, pages 273–288. Springer, 2011.
3. Majid Mohammadi, Wout Hofman, and Yaohua Tan. Simulated annealing-based ontology matching. 2018.
4. Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
5. Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

Wiktionary Matcher

Jan Portisch[1,2][0000−0001−5420−0663], Michael Hladik[2][0000−0002−2204−3138], and Heiko Paulheim[1][0000−0003−4386−8195]

[1] Data and Web Science Group, University of Mannheim, Germany
{jan, heiko}@informatik.uni-mannheim.de
[2] SAP SE Product Engineering Financial Services, Walldorf, Germany
{jan.portisch, michael.hladik}@sap.com

Abstract. In this paper, we introduce *Wiktionary Matcher*, an ontology matching tool that exploits *Wiktionary* as external background knowledge source. *Wiktionary* is a large lexical knowledge resource that is collaboratively built online. Multiple current language versions of *Wiktionary* are merged and used for monolingual ontology matching by exploiting synonymy relations and for multilingual matching by exploiting the translations given in the resource.
We show that *Wiktionary* can be used as external background knowledge source for the task of ontology matching with reasonable matching and runtime performance.[3]

Keywords: Ontology Matching · Ontology Alignment · External Resources · Background Knowledge · Wiktionary

1 Presentation of the System

1.1 State, Purpose, General Statement

The *Wiktionary Matcher* is an element-level, label-based matcher which uses an online lexical resource, namely *Wiktionary*. The latter is "[a] collaborative project run by the Wikimedia Foundation to produce a free and complete dictionary in every language"[4]. The dictionary is organized similarly to Wikipedia: Everybody can contribute to the project and the content is reviewed in a community process. Compared to WordNet [4], *Wiktionary* is significantly larger and also available in other languages than English. This matcher uses *DBnary* [15], an RDF version of *Wiktionary* that is publicly available[5]. The *DBnary* data set makes use of an extended *LEMON* model [11] to describe the data. For this matcher, *DBnary* data sets for 8 *Wiktionary* languages[6] have been downloaded

[4] see https://web.archive.org/web/20190806080601/https://en.wiktionary.org/wiki/Wiktionary
[5] see http://kaiko.getalp.org/about-dbnary/download/
[6] Namely: Dutch, English, French, Italian, German, Portugese, Russian, and Spanish.

and merged into one RDF graph. Triples not required for the matching algorithm, such as glosses, were removed in order to increase the performance of the matcher and to lower its memory requirements. As *Wiktionary* contains translations, this matcher can work on monolingual and multilingual matching tasks. The matcher has been implemented and packaged using the MELT framework[7], a Java framework for matcher development, tuning, evaluation, and packaging [7].

1.2 Specific Techniques Used

Monolingual Matching For monolingual ontologies, the matching system first links labels to concepts in *Wiktionary*, and then checks whether the concepts are synonymous in the external data set. This approach is conceptually similar to an upper ontology matching approach. Concerning the usage of a collaboratively built knowledge source, the approach is similar to *WikiMatch* [5] which exploits the *Wikipedia* search engine.

Wiktionary Matcher adds a correspondence to the final alignment purely based on the synonymy relation independently of the actual word sense. This is done in order to avoid word sense disambiguation on the ontology side but also on *Wiktionary* side: Versions for some countries do not annotate synonyms and translations for senses but rather on the level of the lemma. Hence, many synonyms are given independently of the word sense. In such cases, word-sense-disambiguation would have to be performed also on *Wiktionary* [13].

The linking process is similar to the one presented for the *ALOD2Vec* matching system [14]: In a first step, the full label is looked up on the knowledge source. If the label cannot be found, labels consisting of multiple word tokens are truncated from the right and the process is repeated to check for sub-concepts. This allows to detect long sub-concepts even if the full string cannot be found. Label *conference banquet* of concept *http://ekaw#Conference_Banquet* from the *Conference* track, for example, cannot be linked to the background data set using the full label. However, by applying right-to-left truncation, the label can be linked to two concepts, namely *conference* and *banquet*, and in the following also be matched to the correct concept *http://edas#ConferenceDinner* which is linked in the same fashion.

For multi-linked concepts (such as *conference dinner*), a match is only annotated if every linked component of the label is synonymous to a component in the other label. Therefore, *lens (http://mouse.owl#MA_0000275)* is not mapped to *crystalline_lens (http://human.owl#NCI_C12743)* due to a missing synonymous partner for *crystalline* whereas *urinary bladder neck (http://mouse.owl#MA_0002491)* is matched to *bladder neck (http://human.owl#NCI_C12336)* because *urinary bladder* is synonymous to *bladder*.

Multilingual Matching The multilingual capabilities of the matcher presented in this paper are similar to the work of Lin and Krizhanovsky [10] who use

[7] see `https://github.com/dwslab/melt`

data of the English *Wiktionary* (as of 2010) to allow for multilangual matching of the *COMS* matching system [9]. Unfortunately, the matching system never participated in the OAEI *MultiFarm* track. The work presented here is different in that it uses multiple language versions of *Wiktionary*, the corpora are much larger because they are newer, and in terms of the matching strategy that is applied.

The matcher first determines the language distributions in the ontologies. If the ontologies appear to be in different languages, *Wiktionary* translations are exploited: A match is created, if one label can be translated to the other one according to at least one *Wiktionary* language version – such as the Spanish label *ciudad* and the French label *ville* (both meaning *city*). This process is depicted in figure 1: The Spanish label is linked to the entry in the Spanish *Wiktionary* and from the entry the translation is derived.

If there is no *Wiktionary* version for the languages to be matched or the approach described above yields very few results, it is checked whether the two labels appear as a translation for the same word. The Chinese label 决定 (juédìng), for instance, is matched to the Arabic label قرار (qrār) because both appear as a translation of the English word *decision* on *Wiktionary*. This (less precise) approach is particularly important for language pairs for which no *Wiktionary* data set is available to the matcher (such as Chinese and Arabic). The process is depicted in figure 2: The Arabic and Chinese labels cannot be linked to *Wiktionary* entries but, instead, appear as translation for the same concept.

Instance Matching The matcher presented in this paper can be also used for combined schema and instance matching tasks. If instances are available in the given data sets, the matcher applies a two step strategy: After aligning the schemas, instances are matched using a string index. If there are many instances, *Wiktionary* is not used for the instance matching task in order to increase the matching runtime performance. Moreover, the coverage of schema level concepts in Wiktionary is much higher than for instance level concepts: For example, there is a sophisticated representation of the concept *movie*[8], but hardly any individual movies in Wiktionary.

For correspondences where the instances belong to classes that were matched before, a higher confidence is assigned. If one instance matches multiple other instances, the correspondence is preferred where both their classes were matched before.

Explainability Unlike many other ontology matchers, this matcher uses the extension capabilities of the alignment format [2] in order to provide a human readable explanation of why a correspondence was added to the final alignment. To explain the correspondence involving *(http://cmt_de#c-7914897-1988765, http://conference_en#c-0918067-8070827)*, for instance, the matcher gives the explanation *"The label of entity 1 was found in Wiktionary as 'Konferenz' and translated to 'conference' which equals the normalized label of entity 2."* Such

[8] see `https://en.wiktionary.org/wiki/movie`

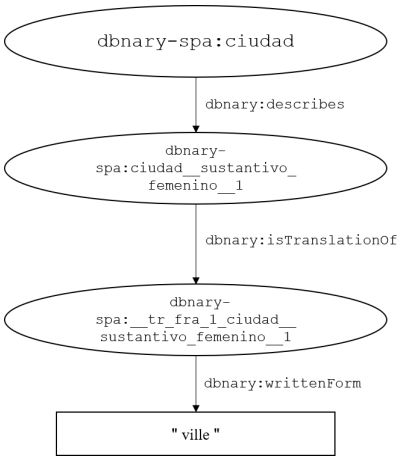


Fig. 1. Translation via the *Wiktionary* headword (using the *DBnary* RDF graph). Here: One (of more) French translations for the Spanish word *ciudad* in the Spanish *Wiktionary*.

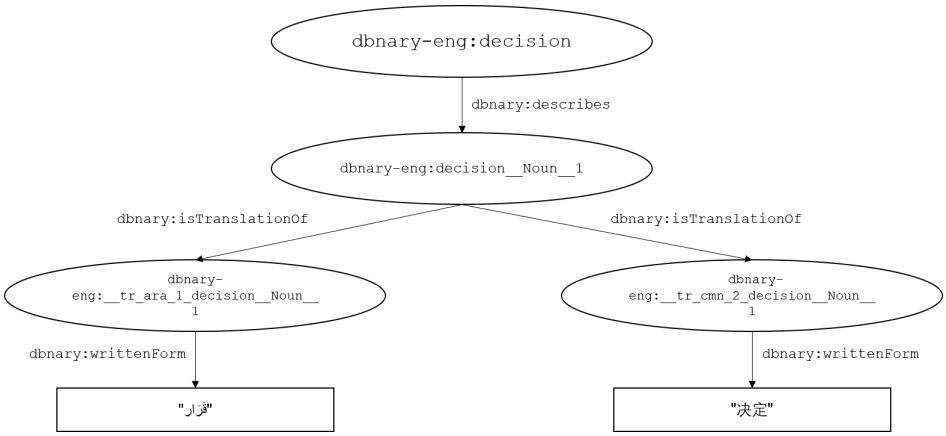


Fig. 2. Translation via the written forms of *Wiktionary* entries (using the *DBnary* RDF graph). Here: An Arabic and a Chinese label appear as translation for the same *Wiktionary* entry (*decision* in the English Wiktionary).

explanations can help to interpret and to trust a matching system's decision. Similarly, explanations also allow to comprehend why a correspondence was falsely added to the final alignment: The explanation for the false positive match *(http://confOf#Contribution, http://iasted#Tax)*, for instance, is given as follows: *"The first concept was mapped to dictionary entry [contribution] and the second concept was mapped to dictionary entry [tax]. According to Wiktionary, those two concepts are synonymous."* Here, it can be seen that the matcher was successful in linking the labels to *Wiktionary* but failed due to the missing word sense disambiguation. In order to explain a correspondence, the `description` property[9] of the *Dublin Core Metadata Initiative* is used.

2 Results

2.1 Anatomy

On the *Anatomy* track [3,1] the matching system achieves a median rank given F_1 scores and significantly outperforms the baseline. The system is capable of finding non-trivial matches such as *temporalis (http://mouse.owl#MA_0002390)* and *temporal muscle (http://human.owl#NCI_C33743)*.

2.2 Conference

The matching system consistently ranks 4th on all reference alignments given F_1 scores in the *Conference* track [16]. Like most matchers, the system achieves better results matching classes compared to matching properties. False positives are in most cases due to string matches and only in some cases due to synonymous relationships such as in (*http://edas#Topic*, *http://iasted#Item*).

2.3 Multifarm

The multilingual approach of the *Wiktionary Matcher* is different from most multilingual ontology matching approaches that use a translation API: Instead of an external function call, multiple multilingual resources are merged and used. Out of the matchers that participated in the *MultiFarm* track [12], *Wiktionary Matcher* performs third with an averaged F_1 score of 0.31 on (i) different ontologies and an averaged F_1 score of 0.12 on (ii) the same but translated ontologies. For the latter task the matching system lacks the ability to recognize that the structure of the ontologies that are to be matched is equal which would be an advantage for this matching problem. As expected, *Wiktionary Matcher* works better for languages for which a data set is available – such as English and French. Compared to other matching systems, the results of this matcher fluctuate more due to missing translation resources for some languages: While the matcher performs competitively for tasks involving the English language, the performance drastically falls when it comes to matching an ontology in the Arabic language.

[9] see `http://purl.org/dc/terms/description`

2.4 Knowledge Graph Track

On the *Knowledge Graph (KG) Track* [8,6], the matcher achieves the second-best result of all submitted matchers on the averaged F_1 scores. Compared to the best matching system, *FCAMap-KG*, the system presented in this paper requires less than a third of the runtime.

The matcher performs better in terms of F_1 on classes and properties compared to instances. This might be due to the fact that the matcher is optimized to match schemas and that the *Wiktionary* background source is only used for the schema matching task.

3 Discussions on the Way to Improve the Proposed System

The current version of *DBnary* does not extract *alternative forms* of words such as *(color, colour)*. This is a limitation by the data set used for this matcher and not by *Wiktionary*. An addition of this relation between lemmas to the data set would likely improve results.

Furthermore, the matching system presented here only uses synonymy and translation relations even though more information is available in the background knowledge source. An extension to other relations that exist between words would help to increase the performance. The false negative match between *intestine secretion* and *intestinal secretion* of classes *http://mouse.owl#MA_0002515* and *http://human.owl#NCI_C32875*, respectively, could be found if the system would exploit the fact that *intestinal* is derived from *intestine* (an information that is available in the data set).

The runtime performance could be improved by loading the background knowledge data (or aggregates) in specialized data structures that allow for a faster data access at runtime, such as key-value stores (rather than querying an RDF graph). This approach could particularly improve the performance on the *MultiFarm* track which has a comparatively slow runtime performance due to complex SPARQL queries.

4 Conclusions

In this paper, we presented the *Wiktionary Matcher*, a matcher utilizing a collaboratively built lexical resource. Given *Wiktionary*'s continuous growth, it can be expected that the matching results will improve over time – for example when additional translations are added. In addition, improvements to the *DBnary* data set, such as the addition of alternative word forms, may also improve the overall matcher performance.

References

1. Bodenreider, O., Hayamizu, T.F., Ringwald, M., de Coronado, S., Zhang, S.: Of mice and men: Aligning mouse and human anatomies.

In: AMIA 2005, American Medical Informatics Association Annual Symposium, Washington, DC, USA, October 22-26, 2005. AMIA (2005), http://knowledge.amia.org/amia-55142-a2005a-1.613296/t-001-1.616182/f-001-1.616183/a-012-1.616655/a-013-1.616652

2. David, J., Euzenat, J., Scharffe, F., dos Santos, C.T.: The alignment API 4.0. Semantic Web **2**(1), 3–10 (2011). https://doi.org/10.3233/SW-2011-0028, https://doi.org/10.3233/SW-2011-0028
3. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., dos Santos, C.T.: Ontology alignment evaluation initiative: Six years of experience. J. Data Semantics **15**, 158–192 (2011). https://doi.org/10.1007/978-3-642-22630-4_6, https://doi.org/10.1007/978-3-642-22630-4_6
4. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Language, Speech, and Communication, MIT Press, Cambridge, Massachusetts (1998)
5. Hertling, S., Paulheim, H.: WikiMatch - Using Wikipedia for Ontology Matching. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N., Stuckenschmidt, H. (eds.) OM-2012: Proceedings of the ISWC Workshop. vol. 946, pp. 37–48 (2012)
6. Hertling, S., Paulheim, H.: Dbkwik: A consolidated knowledge graph from thousands of wikis. In: Wu, X., Ong, Y., Aggarwal, C.C., Chen, H. (eds.) 2018 IEEE International Conference on Big Knowledge, ICBK 2018, Singapore, November 17-18, 2018. pp. 17–24. IEEE Computer Society (2018). https://doi.org/10.1109/ICBK.2018.00011, https://doi.org/10.1109/ICBK.2018.00011
7. Hertling, S., Portisch, J., Paulheim, H.: MELT - Matching EvaLuation Toolkit. In: Semantics 2019 SEM2019 Proceedings. Karlsruhe (2019, to appear)
8. Hofmann, A., Perchani, S., Portisch, J., Hertling, S., Paulheim, H.: Dbkwik: Towards knowledge graph creation from thousands of wikis. In: Nikitina, N., Song, D., Fokoue, A., Haase, P. (eds.) Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017. CEUR Workshop Proceedings, vol. 1963. CEUR-WS.org (2017), http://ceur-ws.org/Vol-1963/paper540.pdf
9. Lin, F., Butters, J., Sandkuhl, K., Ciravegna, F.: Context-based ontology matching: Concept and application cases. In: 10th IEEE International Conference on Computer and Information Technology, CIT 2010, Bradford, West Yorkshire, UK, June 29-July 1, 2010. pp. 1292–1298. IEEE Computer Society (2010). https://doi.org/10.1109/CIT.2010.233, https://doi.org/10.1109/CIT.2010.233
10. Lin, F., Krizhanovsky, A.: Multilingual ontology matching based on wiktionary data accessible via SPARQL endpoint. CoRR **abs/1109.0732** (2011), http://arxiv.org/abs/1109.0732
11. McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., Gracia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D., Wunner, T.: Interchanging Lexical Resources on the Semantic Web. Language Resources and Evaluation **46**(4), 701–719 (Dec 2012). https://doi.org/10.1007/s10579-012-9182-3, http://link.springer.com/10.1007/s10579-012-9182-3
12. Meilicke, C., Garcia-Castro, R., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., Tamilin, A., dos Santos, C.T., Wang, S.: Multifarm: A benchmark for multilingual ontology matching. J. Web Semant. **15**, 62–68 (2012). https://doi.org/10.1016/j.websem.2012.04.001, https://doi.org/10.1016/j.websem.2012.04.001

13. Meyer, C.M., Gurevych, I.: Worth its weight in gold or yet another resource - A comparative study of wiktionary, openthesaurus and germanet. In: Gelbukh, A.F. (ed.) Computational Linguistics and Intelligent Text Processing, 11th International Conference, CICLing 2010, Iasi, Romania, March 21-27, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6008, pp. 38–49. Springer (2010). https://doi.org/10.1007/978-3-642-12116-6_4, https://doi.org/10.1007/978-3-642-12116-6_4
14. Portisch, J., Paulheim, H.: Alod2vec matcher. In: OM@ISWC. CEUR Workshop Proceedings, vol. 2288, pp. 132–137. CEUR-WS.org (2018)
15. Sérasset, G.: Dbnary: Wiktionary as a lemon-based multilingual lexical resource in RDF. Semantic Web **6**(4), 355–361 (2015). https://doi.org/10.3233/SW-140147, https://doi.org/10.3233/SW-140147
16. Zamazal, O., Svátek, V.: The ten-year ontofarm and its fertilization within the onto-sphere. J. Web Semant. **43**, 46–53 (2017). https://doi.org/10.1016/j.websem.2017.01.001, https://doi.org/10.1016/j.websem.2017.01.001

MultiKE: A Multi-view Knowledge Graph Embedding Framework for Entity Alignment*

Wei Hu(✉), Qingheng Zhang, Zequn Sun, and Jiacheng Huang

State Key Laboratory for Novel Software Technology, Nanjing University, China
whu@nju.edu.cn, {qhzhang,zqsun,jchuang}.nju@gmail.com

Abstract. We study the problem of embedding-based entity alignment (EA) between knowledge graphs (KGs), and propose a novel framework that unifies multiple views of entities to learn their embeddings. Experiments on real-world datasets show that this framework largely outperforms the current embedding-based methods.

1 Introduction

Entity alignment (EA) aims to find entities in different knowledge graphs (KGs) referring to the same real-world identity. Conventional methods identify similar entities based on the symbolic features, such as names, textual descriptions and attribute values. Recently, increasing attention has been drawn to leveraging the KG embedding techniques for dealing with this problem, where the key idea is to learn vector representations (called *embeddings*) of KGs and find alignment according to the similarity of the embeddings.

We propose a new EA framework, *MultiKE*, based on multi-view KG embedding. The underlying idea is to divide the various features of KGs into multiple subsets (called *views*), which are complementary to each other (see Figure 1 for example). Thus, entity embeddings can be learnt from each separate view and jointly optimized to improve the alignment performance.

2 Approach

Multi-view KG embedding. Based on the data model of KGs, we define three representative views based on the name, relation and attribute features. First, literals are constituted by sequences of tokens. We embed the name view using the literal embeddings. Second, the relation view characterizes the structure of KGs. We employ TransE to interpret a relation as a translation vector from its head entity to tail entity. Third, for the attribute view, we use a convolutional neural network to extract features from the attributes and values of entities.

Cross-KG training. We propose the cross-KG entity identity inference to capture the alignment information using seed alignment. We also present the cross-KG relation/attribute identity inference to enhance EA.

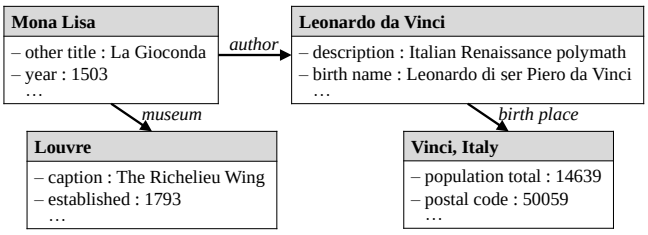


Fig. 1. An example of the multi-view features of four entities in DBpedia. Notations: names (**bold** font), relations (*italic* font) and attributes (regular font).

Table 1. Comparison with existing embedding-based EA methods

Features		Methods	DBP-WD				DBP-YG			
			Hits@1	Hits@10	MR	MRR	Hits@1	Hits@10	MR	MRR
Rel. +	Attributes	JAPE [2]	31.84	58.88	266	0.411	23.57	48.41	189	0.320
Rel. +	Textual desc.	KDCoE [1]	57.19	69.53	182	0.618	42.71	48.30	137	0.446
Rel. +	Literals	AttrE [4]	38.96	66.77	142	0.487	23.24	42.70	706	0.300
Multiple views		MultiKE-WVA	90.42	94.59	22	0.921	85.92	94.99	19	0.891
Multiple views		MultiKE-SSL	91.86	96.26	39	0.935	82.35	93.30	21	0.862
Multiple views		MultiKE-ITC	91.45	95.19	114	0.928	88.03	95.32	35	0.906

View combinations. Intuitively, general entity embeddings can benefit from multiple view-specific embeddings. We propose weighted view averaging (WVA), shared space learning (SSL) and in-training combination (ITC).

3 Evaluation

We selected two datasets in [3], DBP-WD and DBP-YG, and compared MultiKE with JAPE, KDCoE and AttrE, each of which used one type of extra features as enhancement. Table 1 shows that MultiKE largely outperformed the others.

4 Conclusion

In this paper, we proposed a multi-view KG embedding framework for EA, and our experiments demonstrated its effectiveness. In future work, we will investigate more feasible views (e.g., entity types) and study cross-lingual EA.

References

1. Chen, M., Tian, Y., Chang, K.W., Skiena, S., Zaniolo, C.: Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In: IJCAI. pp. 3998–4004 (2018)
2. Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding. In: ISWC. pp. 628–644 (2017)
3. Sun, Z., Hu, W., Zhang, Q., Qu, Y.: Bootstrapping entity alignment with knowledge graph embedding. In: IJCAI. pp. 4396–4402 (2018)
4. Trsedya, B.D., Qi, J., Zhang, R.: Entity alignment between knowledge graphs using attribute embeddings. In: AAAI (2019)

MTab: Matching Tabular Data to Knowledge Graph with Probability Models

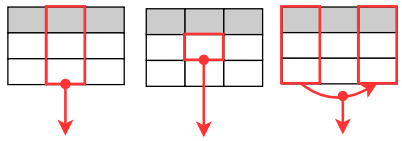
Phuc Nguyen[1,2], Natthawut Kertkeidkachorn[3], Ryutaro Ichise[1,2,3], and Hideaki Takeda[1,2]

[1] National Institute of Informatics, Japan
[2] SOKENDAI (The Graduate University for Advanced Studies), Japan
[3] National Institute of Advanced Industrial Science and Technology, Japan

Abstract. This paper presents the design of our system, namely MTab, for Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2019). MTab combines the voting algorithm and the probability model to solve critical bottlenecks of the matching task. Results on SemTab 2019 show MTab obtains the promising performance.

1 Introduction

Tabular Data to Knowledge Graph Matching (SemTab 2019) [4] is a challenge on matching semantic tags from table elements to knowledge bases (KBs), especially DBpedia. Fig. 1 depicts the three sub-tasks for SemTab 2019. Given a table data, **CTA** (Fig. 1a) is the task of assigning a semantic type (e.g., a DBpedia class) to a column. In **CEA** (Fig. 1b), a cell is linked to an entity in KB. The relation between two columns is assigned to a property in KB in **CPA** (Fig. 1c).



(a) **CTA** (b) **CEA** (c) **CPA**

Fig. 1. Tabular Data Matching to Knowledge Base (DBpedia)

2 Approach

To address the three tasks of the challenge, we designed our system (MTab) by the 4-steps pipeline as shown in Fig. 2.

Step 1 is to pre-process a table data by predicting languages of the table with fasttext [1], correcting spelling, predicting data types (e.g., number or text), and searching relevant entities in DBpedia. Due to the heterogeneous problem, we utilize entity searching on many services including DBpedia Lookup, DBpedia

[4] http://www.cs.ox.ac.uk/isg/challenges/sem-tab/

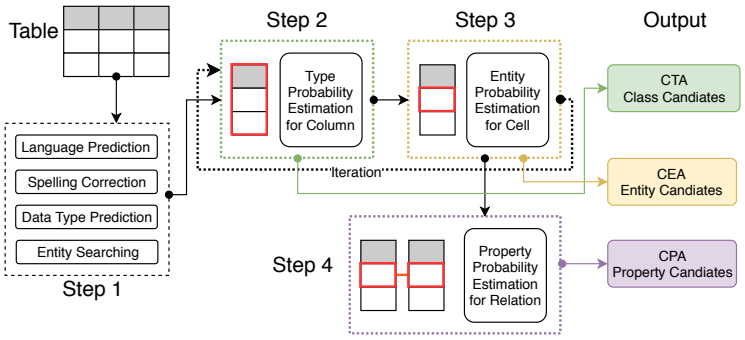


Fig. 2. The design of MTab framework

endpoint. Also, we search relevant entities on Wikipedia and Wikidata by redirected links to DBpedia to increase the possibility of finding the relevant entities. We assume that cells in a column have the same type. We then use information from Step 1 to estimate the probability of the types for the column in Step 2. The type candidate which has the highest probability is the result for **CTA** task. In Step 3, the result of Step 2 and information of Step 1 are used to estimate the probability for entities. Similarly, the entity candidate which has the highest probability is the result for **CEA** task. In Step 4, we use the result from Step 3 to estimate the property between two entities, and then, adopt the voting technique to estimate the probability for all rows of two columns. The result for **CPA** is the highest probability of property candidate in Step 4. We repeatedly execute Step 2, 3 and 4 to find the best candidates for columns, cells, and the relation between two columns.

3 Results and Conclusion

Table 1 reports the overall results of MTab for three matching tasks. Overall, these results show that MTab achieves a promising performance for the three Tabular data matching tasks. The MTab performance might be explained in part by searching cell values from multiple services to increase the possibility of finding the relevant entities, and adopting the iteration procedure to boost the overall performance for the three tasks.

Table 1. Results of MTab on Round 1 Data of SemTab 2019

Task	F1	Precision	Recall
CEA	0.816	0.799	0.834
CTA	0.934	0.926	0.942
CPA	0.594	0.698	0.516

References

1. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: EACL 2017. pp. 427–431. ACL (April 2017)

Generating Referring Expressions from Knowledge Graphs

Armita Khajeh Nassiri[2], Nathalie Pernelle[1], and Fatiha Saïs[1]

[1] LRI, Paris Sud University, CNRS 8623, Paris Saclay University, Orsay F-91405, France
firstname.lastname@lri.fr
[2] École Polytechnique, Palaiseau F-91128, France
armita.khajeh-nassiri@polytechnique.edu

1 Introduction

A *referring expression* (RE) is a description in natural language or a logical formula that can uniquely identify an entity. For instance, the 44th president of the United States unambiguously characterizes Barack Obama. Referring expressions find applications in disambiguation, data anonymization, query answering, or data linking. There may potentially exist many logical expressions for uniquely identifying an entity. Generation of referring expressions is a well-studied task in natural language generation [1]. Hence, various algorithms with different objectives have been proposed to automatically discover REs. These approaches vary depending on the expressivity of the logical formulas they can generate. For instance in [1, 2], REs that are created are conjunctions of atoms. While in [3], more complex REs represented in description logics are discovered that can involve the universal quantifier.
In this work our focus lies on automatically discovering REs for each entity within a class of a knowledge graph. *Keys* of a class are sets of properties whose values can uniquely identify one entity of that class. Hence, if the properties for the keys are instantiated, they can each be considered as a referring expression. What interests us in this work, is to efficiently discover REs by focusing on the ones that cannot be found by instantiating the keys. It should be noted that the quality of REs we discover is very dependent on the dataset. The completeness, correctness and lack of noise in the knowledge graph plays a pivotal role in how good and interpretable REs are.

2 Referring Expression Generation Approach

In this work, we discover **minimal** REs existing in a class. By minimality, we mean that there is no other RE that we discover and that can be logically entailed by the minimal one. The REs we mine always consist of conjunctions that specify the classes the entities belong to.

To generate REs for a given class C, we start by creating the maximal non-keys of C (the set of properties such that addition of a property will make it a key for that class) using SAKey [4]. The algorithm first generates candidate expressions containing one instantiated property (i.e. $p(x, v)$). Whenever an expression E only describes one instance i of C, E is output as a referring expression. Adding more properties to the

description E will still uniquely identify the instance i, just making it more complex. Hence, we remove the REs (e.g. $p(i, v)$) found at the end of this step and reduce the search space. Then, the remaining candidate expressions are taken into account with one more property at each step, until either the search space is empty or there is no more set of non-keys to consider. To increase the depth of subgraph, we have to consider the class of the new individual and obtain its corresponding set of maximal non-keys so that the process can be reiterated. Some pruning techniques can be applied to limit the size and the complexity of the REs discovered by our approach. For instance the depth of the graph pattern and number of allowed variables can be limited.

3 Experimental Evaluation

We chose YAGO as the knowledge graph on which we discover the REs and used 10 different classes such as Actor, City and Book (same data used in VICKEY [5]). We mined REs of depth one and for example, for the class City (with 1.1M triples) we found 1.2M REs in less than 2 minutes. On average, our approach can detect from 1.5 to 14.3 RE per individual depending on the class.
This approach can discover RE such as: *made in heaven* is the album created by Queen in the year 1991. Among the actors, only *George Clooney* has been born in Lexington-Kentucky in the year 1961. When we ran the algorithm with depth 2, we obtained REs like *Alfred Werner* is a scientist who has won the Nobel Prize in Chemistry and has graduated from a university located in Zurich.

4 Conclusion

In this paper, we proposed an approach that can efficiently discover REs by reducing the search space thanks to maximal non-keys. Due to the incompleteness of knowledge graphs, entity linking using keys may be insufficient to link all individuals. We expect that using REs will increase the recall of rule-based data linking methods.

References

1. R. Dale. *Generating referring expressions - constructing descriptions in a domain of objects and processes*. ACL-MIT press series in natural language processing. MIT Press, 1992.
2. E. Krahmer, S. v. Erk, and A. Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72, 2003.
3. Y. Ren, J. Z. Pan, and Y. Zhao. Towards soundness preserving approximation for abox reasoning of OWL2. In *Proceedings of the 23rd International Workshop on Description Logics (DL 2010), Waterloo, Ontario, Canada, May 4-7, 2010*, 2010.
4. D. Symeonidou, V. Armant, N. Pernelle, and F. Saïs. SAKey: Scalable Almost Key Discovery in RDF Data. In S. Verlag, editor, *In proceedings of the 13th International Semantic Web Conference, ISWC 2014*, volume Lecture Notes in Computer Science, pages 33–49, Riva del Garda, Italy, Oct. 2014.
5. D. Symeonidou, L. Galàrraga, N. Pernelle, F. Saïs, and F. Suchanek. VICKEY: Mining Conditional Keys on Knowledge Bases. In *International Semantic Web Conference (ISWC)*, volume 10587 of *Lecture Notes in Computer Science*, pages 661–677, Austria, Oct 2017. Springer.

Semantic Table Interpretation using MantisTable

Marco Cremaschi[1], Anisa Rula[1,2], Alessandra Siano[1], and Flavio De Paoli[1]

[1] University of Milano - Bicocca, Italy
{marco.cremaschi,anisa.rula,flavio.depaoli}@unimib.it
a.siano2@campus.unimib.it
[2] Univeristy of Bonn, Germany
{rula}@cs.uni-bonn.de

Keywords: Knowledge Graph · Semantic Interpretation · Table Annotation

1 Introduction & Motivation

A vast amount of relevant structured data represented in tables are available on the Web. However, querying such data is difficult since they are incorporated in HTML web pages and are not easily query-able. Some approaches started to propose [4, 2] extraction, annotation and transformation of tabular data into machine-readable formats. The problem of annotating tables also known as *Semantic Table Interpretation (STI)* takes a relational table and a Knowledge Graph (KG) in input, and returns a semantically annotated table in output [1]. In this paper, we propose *MantisTable*[3], a web interface and an open source Semantic Table Interpretation tool that automatically annotates, manages and makes accessible to humans and machines the semantic of tables. Although STI contains several steps the key feature of our tool is the involvement of all the STI steps that run fully automatically.

2 Overview of MantisTable

Figure 1 shows the architecture of MantisTable which is designed to be modular:

View Layer provides a graphic user interface to serve different types of tasks such as storing and loading tables, exploration of the annotated tables which allow users to navigate all the executed steps by clicking on each phase and analyse the result, execution of the STI steps and the editing which allow users to understand what has been achieved and give them the opportunity to modify and enhance the results.

Controller Layer creates all the abstraction between the View layer and the Model layer and implements all the STI steps as follows:

Data Preparation cleans and normalizes values in the table. Transformations applied to tables include text normalization such as solve acronyms and abbreviations by applying regular expressions [3]. **Column Analysis** assigns types

[3] http://mantistable.disco.unimib.it

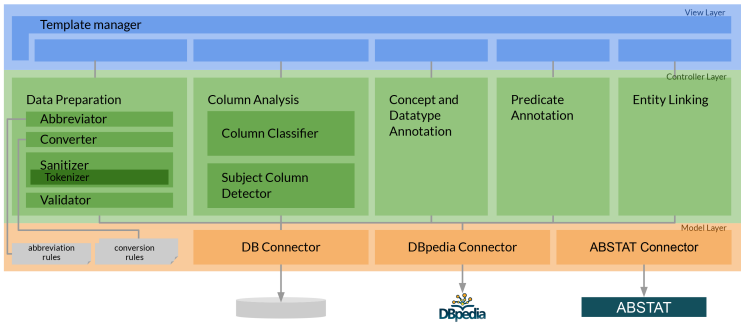


Fig. 1. Architecture of MantisTable tool.

to columns that are named entity (NE-column) or literal column (L-column), and then identify the subject column (S-column). To identify L-column candidates the tool considers 16 regular expressions that identify several Regextypes. If the number of occurrences of the most frequent Regextype in a column exceeds a given threshold, that column is annotated as L-column, otherwise, it is annotated as NE-column. To detect the S-column, the tool considers the NE-columns on which applies different statistic features (e.g. % of cells with unique content). **Concept and Datatype Annotation** identifies the mappings between columns headers and semantic elements (concepts or datatypes) in a KG. First, we perform the entity-linking by searching the KG with the content of a cell, to get a set of candidate entities and use the DICE similarity measure for text disambiguation. Second, the abstract and all concepts for each winning entity are retrieved from DBpedia. For each extracted concept, we count the occurrences in the abstract. For the Datatype Annotation we consider the L-columns and for the identification of datatypes, a Regextype is applied on the content of each column. **Predicate Annotation** finds relations, in the form of predicates, between the S-column and the object columns to set the overall meaning of the table. Further, the entities identified as subjects and objects are searched in the KG to identify the correct predicate. **Entity Linking** deals with mappings between the content of cells and entities in the KG.

Model Layer considers mainly data access for communicating with an application's data sources such as DB connector or DBpedia connector.

References

1. Cremaschi, M., Rula, A., Siano, A., De Paoli, F.: MantisTable: a Tool for Creating Semantic Annotations on Tabular Data. In: 16th ESWC: Posters and Demos (2019)
2. Pham, M., Alse, S., Knoblock, C.A., Szekely, P.: Semantic labeling: a domain-independent approach. In: 15th ISWC (2016)
3. Ritze, D., Lehmberg, O., Bizer, C.: Matching HTML Tables to DBpedia. In: WIMS (2015)
4. Zhang, Z.: Effective and efficient semantic table interpretation using tableminer+. Semantic Web (2017)

Towards explainable entity matching via comparison queries

Alina Petrova, Egor V. Kostylev, Bernardo Cuenca Grau, and Ian Horrocks

Department of Computer Science, University of Oxford
{alina.petrova, egor.kostylev, bernardo.cuenca.grau, ian.horrocks}@cs.ox.ac.uk

Nowadays there exists an abundance of heterogeneous Semantic Web data coming from multiple sources. As a result, matching Linked Data has become a tedious and non-transparent task. One way to facilitate entity matching across datasets is to provide human-readable explanations that highlight what the two entities have in common, as well as what differentiates the two entities.

Entity comparison is an important information exploration problem that has recently gained considerable research attention [1, 2, 4]. In this paper we propose a solution towards explainable entity matching in Linked Data where entity comparison is used as a subroutine that assists in debugging and validation of matchings. To this end, we adopt the entity comparison framework in which explanations are modelled as unary conjunctive queries of restricted form [3, 4].

We concentrate on the data model where a *dataset* is an RDF graph—that is, a set of triples of IRIs and literals, jointly called *entities*. The basic building block of a query is a *triple pattern*, which is a triple of entities and variables. Then, a *query* is a non-empty finite set of triple patterns in which one variable, usually denoted by X, is an *answer variable*. The set $Q(D)$ of *answer* entities to a query Q on a dataset D is defined as usual in databases.

The two main notions of the framework are the similarity and difference queries for pairs of entities, which are defined as follows: a *similarity query* for entities a and b in a dataset D is a query Q satisfying $\{a, b\} \subseteq Q(D)$; a *difference query* for a relative to b is a query Q satisfying $a \in Q(D)$ and $b \notin Q(D)$.

In our prior work we proposed an algorithm for computing comparison queries that can be repurposed for similarity and difference queries [3]. The algorithm is based on the computation of a *similarity tree*—a data structure that represents commonalities and discrepancies in data for input entities a and b. It is a directed rooted tree with nodes and edges labelled by pairs of sets of entities such that the root is labelled by $(\{a\}, \{b\})$ and every edge labelled (E_1, E_2) between nodes labelled (N_1, N_2) and (N_1', N_2') is justified in the sense that for every entity n in N_i, $i \in \{1, 2\}$, there is a triple (n, e, n') in the dataset with $e \in E_i$ and $n' \in N_i'$.

For instance, suppose there are 3 entities, Emma_Watson, Emily_Watson and E_Watson, that need to be either matched or disambiguated, and a data fragment given in Figure 1. Then the similarity trees for Emma_Watson and E_Watson, and for Emily_Watson and E_Watson are depicted in Figure 2 (where singleton sets $\{\ell\}$ and pairs $(\{\ell\}, \{\ell\})$ are both written as ℓ for readability).

```
Emma_Watson:                        E_Watson:                        Emily_Watson:

Emma_Watson nationality British     E_Watson actedIn Ballet_Shoes    Emily_Watson nationality British
Emma_Watson actedIn Harry_Potter    E_Watson actedIn Little_Women    Emily_Watson actedIn Little_Women1
Emma_Watson actedIn Ballet_Shoes    Little_Women year 2019           Little_Women1 year 2017
```

Fig. 1. A fragment of data involving three concepts to be matched

Each branch in a similarity tree can be treated as a separate similarity query, in which each edge is encoded as a triple pattern, and each label (L_1, L_2) is encoded as either an entity ℓ, if $L_1 = L_2 = \{\ell\}$) or a fresh variable otherwise. For example, a query $Q_1 = (X, \mathsf{actedIn}, \mathsf{Ballet\_Shoes})$ is a similarity query for $\mathsf{Emma\_Watson}$ and $\mathsf{E\_Watson}$, while a query $Q_2 = (X, \mathsf{actedIn}, Y), (Y, \mathsf{year}, Z)$ is a similarity query for $\mathsf{Emily\_Watson}$ and $\mathsf{E\_Watson}$. Moreover, each branch involving non-entity labels can also be treated as a difference query, if instead of some variables we take entities from one of the label sets. For example, query

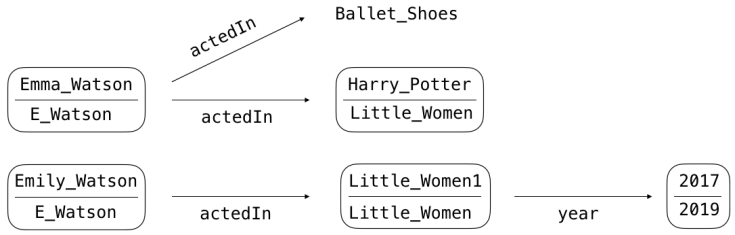


Fig. 2. Similarity trees rooted in two pairs of entities

$Q_3 = (X, \mathsf{actedIn}, \mathsf{Little\_Women}), (\mathsf{Little\_Women}, \mathsf{year}, 2019)$ is a difference query for $\mathsf{E\_Watson}$ relative to $\mathsf{Emily\_Watson}$.

Both types of queries can assist in explaining why two entities should or should not be merged: Q_1 gives a good reason to match $\mathsf{Emma\_Watson}$ and $\mathsf{E\_Watson}$ into one entity, Q_2 is not specific enough to match the other pair, and Q_3 can act as an indicator that the two movies named $\mathsf{Little\_Women}$ are indeed two different movies, and $\mathsf{Emily\_Watson}$ and $\mathsf{E\_Watson}$ are different people.

References

1. Colucci, S., Giannini, S., Donini, F.M., Di Sciascio, E.: Finding commonalities in Linked Open Data. In: Proc. of CILC. pp. 324–329 (2014)
2. El Hassad, S., Goasdoué, F., Jaudoin, H.: Learning commonalities in SPARQL. In: Proc. of ISWC. pp. 278–295 (2017)
3. Petrova, A., Kostylev, E.V., Cuenca Grau, B., Horrocks, I.: Query-based entity comparison in knowledge graphs revisited. In: Proc. of ISWC (2019)
4. Petrova, A., Sherkhonov, E., Cuenca Grau, B., Horrocks, I.: Entity comparison in RDF graphs. In: Proc. of ISWC. pp. 526–541 (2017)

Discovering Expressive Rules for Complex Ontology Matching and Data Interlinking

Manuel Atencia[1], Jérôme David[1], Jérôme Euzenat[1], Liliana Ibanescu[2], Nathalie Pernelle[3], Fatiha Saïs[3], Élodie Thiéblin[4], and Cassia Trojahn[4]

[1] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
firstname.lastname@inria.fr
[2] UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay, 75005 Paris, France
firstname.lastname@agroparistech.fr
[3] LRI, Paris Sud University, CNRS 8623, Paris Saclay University, Orsay F-91405, France
firstname.lastname@lri.fr
[4] IRIT, UMR 5505, 1118 Route de Narbonne, F-31062 Toulouse, France
firstname.lastname@irit.fr

1 Introduction

Ontology matching and data interlinking as distinct tasks aim at facilitating the interoperability between different knowledge bases. Although the field has fully developed in the last years, most ontology matching works still focus on generating simple correspondences (e.g., $Author \equiv Writer$). These correspondences are however insufficient to fully cover the different types of heterogeneity between knowledge bases and complex correspondences are required (e.g., $LRIMember \equiv Researcher \sqcap \exists belongsToLab.\{LRI\}$). Few approaches have been proposed for generating complex alignments, focusing on correspondence patterns or exploiting common instances between the ontologies. Similarly, unsupervised data interlinking approaches (which do not require labelled samples) have recently been developed. One approach consists in discovering linking rules on unlabelled data, such as simple keys [2] (e.g., $\{lastName, lab\}$) or conditional keys [3] (e.g., $\{lastName\}$ under the condition $c = Researcher \sqcap \exists lab.\{LRI\}$). Results have shown that the more expressive the rules are, the higher the recall is. However naive approaches cannot be applied on large datasets. Existing approaches presuppose either that the data conform to the same ontology [2] or that all possible pairs of properties be examined [1]. Complementary, link keys are a set of pairs of properties that identify the instances of two classes of two RDF datasets [1] (e.g., $\{\langle creator, auteur\rangle, \langle title, titre\rangle\}$ linkkey $\langle Book, Livre\rangle$, expresses that instances of the $Book$ class which have the same values for properties $creator$ and $title$ as an instance of the $Livre$ class has for $auteur$ and $titre$ are the same). Such, link keys may be directly extracted without the need for an alignment.

2 Proposed approach

We introduce here an approach that aims at evaluating the impact of complex correspondences in the task of data interlinking established from the application of keys (Figure 1).Given two populated ontologies O_1 and O_2, we first apply the CANARD system [4]

for establishing complex correspondences (1). Then, the key discovery tools VICKEY [3] and LinkEx are applied for the discovery of simple keys, conditional keys, and link keys from the instances of O_1 and O_2, exploiting the complex correspondences as input (as a way of reducing the key search space) (2). The keys are then applied in the data interlinking task, which can also benefit from the complex correspondences (as a way of extending the sets of instances to be compared) (3). Finally, as CANARD considers shared instances, the matching is iterated by considering the detected identity links.

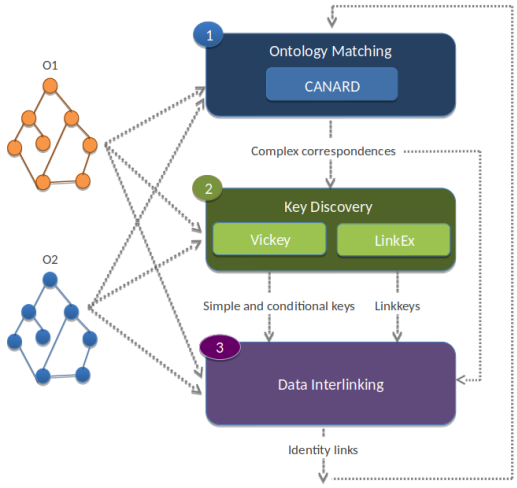


Fig. 1. Workflow of ontology matching and data interlinking enhanced by key discovery.

We plan to evaluate the approach to verify, on the one hand, whether the use of complex correspondences allows to improve the results of data interlinking. On the other hand, thanks to the use of the detected identity links, it would also be reasonable to expect improvements in ontology matching results. Experiments will be run on DBpedia and YAGO, covering different domains such as people, organizations, and locations, as there exists reference entity links or these datasets.

Acknowledgement. This work is supported by the CNRS Blanc project RegleX-LD.

References

1. M. Atencia, J. David, and J. Euzenat. Data interlinking through robust linkkey extraction. In *ECAI*, pages 15–20, 2014.
2. D. Symeonidou, V. Armant, N. Pernelle, and F. Saïs. Sakey: Scalable almost key discovery in rdf data. In *ISWC*, pages 33–49, 2014.
3. D. Symeonidou, L. Galárraga, N. Pernelle, F. Saïs, and F. M. Suchanek. VICKEY: mining conditional keys on knowledge bases. In *ISWC*, pages 661–677, 2017.
4. É. Thiéblin, O. Haemmerlé, and C. Trojahn. CANARD complex matching system: results of the 2018 OAEI evaluation campaign. In *OM@ISWC*, pages 138–143, 2018.

Decentralized Reasoning on a Network of Aligned Ontologies with Link Keys

Jérémy Lhez[1], Chan Le Duc[1], Thinh Dong[2], and Myriam Lamolle[1]

[1] LIASD, Université Paris 8 - IUT de Montreuil, France
{lhez,leduc,lamolle}@iut.univ-paris8.fr
[2] University of Danang, Vietnam
dnnthinh@kontum.udn.vn

1 Introduction

Reasoning on a network of aligned ontologies has been investigated in different contexts where the semantics given to correspondences differs from one to another. In this paper, we introduce a new semantics of correspondences which is weaker than the usual one and propose a procedure for reasoning over a network of aligned ontologies with link keys [1] in a decentralized manner, i.e. reasoning can be independently performed on different sites This process allows to reduce polynomially global reasoning to local reasoning.

To achieve such results for a network of ontologies expressed in the description logic $\mathcal{ALC}$, the semantics of a correspondence, denoted $C \rightarrow D$ where C and D are concepts in ontologies O_i and O_j respectively, is defined as an implication of concept unsatisfiabilities (i.e. unsatisfiability of D implies unsatisfiability of C) rather than a concept subsumption as usual. This weakened semantics allows to reduce the reasoning complexity over a network of aligned ontologies since (i) only individual equalities and concept unsatisfiabilities such as $a \approx b$, $C \sqsubseteq \bot$ can be propagated from one to another ontology, and (ii) if a concept is locally unsatifiable in an ontology then it remains unsatisfiable when adding to the ontology individual equalites or concept unsatisfiabilities. The weakened semantics would be relevant for correspondences between ontologies of different nature. Given two ontologies about `equipment` and `staff` and a correspondence $\mathsf{Computer} \rightarrow \mathsf{Developer}$ between them. With this correspondence, the weakened semantics tells us that if there is no developer then there is no computer. The standard semantics is irrelevant in this case.

We use $\langle \{O_i\}_{i=1}^n, \{A_{ij}\}_{i=1,j=2,i<j}^n \rangle$ to denote a network of ontologies where each O_i is an ontology expressed in $\mathcal{ALC}$ and each A_{ij} contains individual correspondences, link keys with the usual semantics, or concept correspondences with the weakened semantics. Such a network is *consistent* if there is a model $\mathcal{I}_i$ of each ontology O_i which satisfies all correspondences in each A_{ij}. We will present our algorithms for a network composed of two ontologies O_1, O_2 and an alignment A_{12}. These algorithms can be straightforwardly extended to a general network of aligned ontologies.

2 Decentralized reasoning by propagating

As discussed in Section 1, the weakened semantics of alignments allows us to decompose checking consistency of a network into two steps which consist in propagating knowledge from one to another ontology.

Propagating individual equalities. This step discovers all inferred individual equalities by applying link keys in the alignment and using local reasoners for entailment on ontologies. For example, a new individual correspondence $a \approx b$ will be discovered and added to A_{12} if $\{\langle P, Q\rangle\}\mathsf{linkkey}\langle C, D\rangle \in A_{12}$, $c \approx d \in A_{12}$ and $P(a, c) \in O_1$, $Q(b, d) \in O_2$. A new equality $a \approx c$ will be discovered and added to O_1 if $a \approx b, c \approx b \in A_{12}$. When a local reasoner is called to check whether $O_i \models a \approx b$, it needs only one O_i for reasoning.

Propagating concept unsatisfiabilities. This step uses local reasoners associated with ontologies to discover from each ontology O_i new unsatisfiable concepts which can result from unsatisfiable concepts in O_j via concept correspondences. For instance, if $O_1 \models C \sqsubseteq \bot$ and $C \leftarrow D \in A_{12}$ then a new axiom $D \sqsubseteq \bot$ will be added to O_2. As the previous step, each local reasoner needs only one O_i for reasoning.

The main algorithm executes these two steps until either an inconsistency is found, or a stationary state is reached. If O_1, O_2 are consistent, and A_{12} does not contain any pair $a \approx b, a \not\approx b$, then the network itself is consistent. Our algorithm runs in polynomial time in the size of the network since the propagation procedures add only axioms and assertions which are composed of (sub-)concepts and named individuals occurring in the ontologies and alignments. Moreover, these algorithms never remove anything from the network.

Implementation and tests. The algorithms has been implemented and integrated within DRAOn [2]. HermiT [3] is used for local reasoners. We performed some tests with several datasets available from the OAEI web site. We compared the performances of DRAOn under the IDDL semantics [2] and the weakened semantics. Better performance has been observed for the latter. For instance, checking consistency of the network composed of SNOMED, FMA and the alignment took 81 seconds under the weakened semantics while it took greater than 15 minutes under the IDDL semantics. We also added to the alignments some link keys, and ran other tests to validate the implementation of our algorithm.

References

1. Gmati, M., Atencia, M., Euzenat, J.: Tableau extensions for reasoning with link keys. In: Proceedings of the 11th International Workshop on Ontology Matching. (2016) 37–48
2. Le Duc, C., Lamolle, M., Zimmermann, A., Curé, O.: Draon: A distributed reasoner for aligned ontologies. In: Informal Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE). (2013) 81–86
3. Shearer, R., Motik, B., Horrocks, I.: HermiT: A Highly-Efficient OWL Reasoner. In: Proc. of the 5th Int. Workshop on OWL (OWLED). (2008)